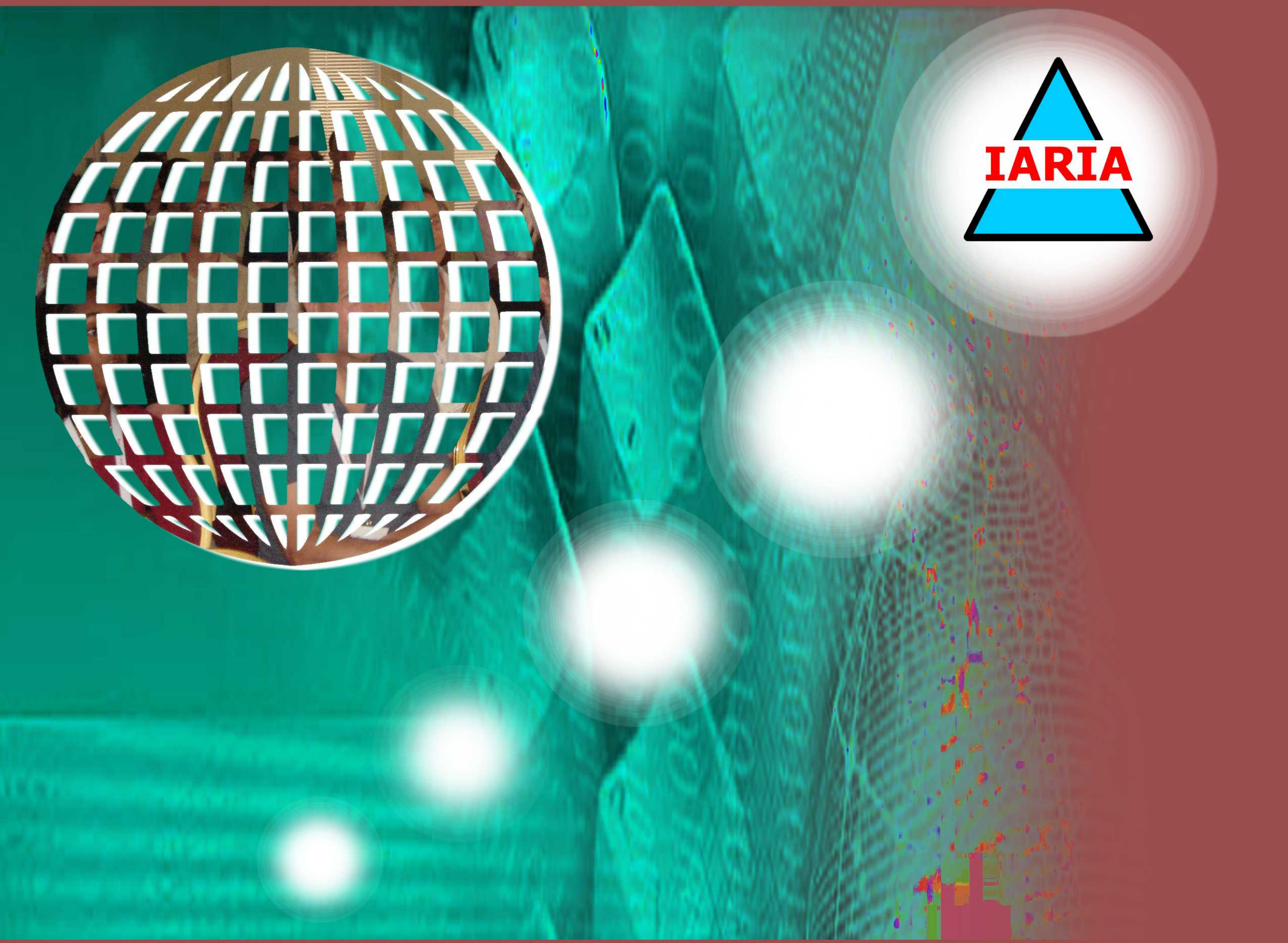
International Journal on Advances in Software



2020 vol. 13 nr. 3&4

The *International Journal on Advances in Software* is published by IARIA.
ISSN: 1942-2628
journals site: http://www.iariajournals.org
contact: petre@iaria.org

International Journal on Advances in Software
Volume 13, Number 3 & 4, 2020

Peter Baumann, Jacobs University Bremen / Rasdaman GmbH Bremen, Germany
Gabriele Bavota, University of Salerno, Italy
Grigorios N. Beligiannis, University of Western Greece, Greece
Noureddine Belkhatir, University of Grenoble, France
Jorge Bernardino, ISEC - Institute Polytechnic of Coimbra, Portugal
Rudolf Berrendorf, Bonn-Rhein-Sieg University of Applied Sciences - Sankt Augustin, Germany
Ateet Bhalla, Independent Consultant, India
Fernando Boronat Seguí, Universidad Politecnica de Valencia, Spain
Pierre Borne, Ecole Centrale de Lille, France
Farid Bourennani, University of Ontario Institute of Technology (UOIT), Canada
Narhimene Boustia, Saad Dahlab University - Blida, Algeria
Hongyu Pei Breivold, ABB Corporate Research, Sweden
Carsten Brockmann, Universität Potsdam, Germany
Antonio Bucchiarone, Fondazione Bruno Kessler, Italy
Georg Buchgeher, Software Competence Center Hagenberg GmbH, Austria
Dumitru Burdescu, University of Craiova, Romania
Martine Cadot, University of Nancy / LORIA, France
Isabel Candal-Vicente, Universidad Ana G. Méndez, Puerto Rico
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Jose Carlos Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Alain Casali, Aix-Marseille University, France
Yaser Chaaban, Leibniz University of Hanover, Germany
Savvas A. Chatzichristofis, Democritus University of Thrace, Greece
Antonin Chazalet, Orange, France
Jiann-Liang Chen, National Dong Hwa University, China
Shiping Chen, CSIRO ICT Centre, Australia
Wen-Shiung Chen, National Chi Nan University, Taiwan
Zhe Chen, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China PR
Po-Hsun Cheng, National Kaohsiung Normal University, Taiwan
Yoonsik Cheon, The University of Texas at El Paso, USA
Lau Cheuk Lung, INE/UFSC, Brazil
Robert Chew, Lien Centre for Social Innovation, Singapore
Andrew Connor, Auckland University of Technology, New Zealand
Rebeca Cortázar, University of Deusto, Spain
Noël Crespi, Institut Telecom, Telecom SudParis, France
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Duilio Curcio, University of Calabria, Italy
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Paulo Asterio de Castro Guerra, Tapijara Programação de Sistemas Ltda. - Lambari, Brazil
Cláudio de Souza Baptista, University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autonónoma de México, México
Rafael del Vado Vírseda, Universidad Complutense de Madrid, Spain
Giovanni Denaro, University of Milano-Bicocca, Italy
Nirmit Desai, IBM Research, India
Vincenzo Deufemia, Università di Salerno, Italy
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Javier Diaz, Rutgers University, USA
Nicholas John Dingle, University of Manchester, UK
Roland Dodd, CQUniversity, Australia
Aijuan Dong, Hood College, USA
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Cédric du Mouza, CNAM, France

Ann Dunkin, Palo Alto Unified School District, USA
Jana Dvorakova, Comenius University, Slovakia
Hans-Dieter Ehrich, Technische Universität Braunschweig, Germany
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Atilla Elçi, Aksaray University, Turkey
Khaled El-Fakih, American University of Sharjah, UAE
Gledson Elias, Federal University of Paraíba, Brazil
Sameh Elnikety, Microsoft Research, USA
Fausto Fasano, University of Molise, Italy
Michael Felderer, University of Innsbruck, Austria
João M. Fernandes, Universidade de Minho, Portugal
Luis Fernandez-Sanz, University of de Alcala, Spain
Felipe Ferraz, C.E.S.A.R, Brazil
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Wolfgang Fohl, Hamburg Universiy, Germany
Simon Fong, University of Macau, Macau SAR
Gianluca Franchino, Scuola Superiore Sant'Anna, Pisa, Italy
Naoki Fukuta, Shizuoka University, Japan
Martin Gaedke, Chemnitz University of Technology, Germany
Félix J. García Clemente, University of Murcia, Spain
José García-Fanjul, University of Oviedo, Spain
Felipe Garcia-Sanchez, Universidad Politecnica de Cartagena (UPCT), Spain
Michael Gebhart, Gebhart Quality Analysis (QA) 82, Germany
Tejas R. Gandhi, Virtua Health-Marlton, USA
Andrea Giachetti, Università degli Studi di Verona, Italy
Afzal Godil, National Institute of Standards and Technology, USA
Luis Gomes, Universidade Nova Lisboa, Portugal
Diego Gonzalez Aguilera, University of Salamanca - Avila, Spain
Pascual Gonzalez, University of Castilla-La Mancha, Spain
Björn Gottfried, University of Bremen, Germany
Victor Govindaswamy, Texas A&M University, USA
Gregor Grambow, AristaFlow GmbH, Germany
Carlos Granell, European Commission / Joint Research Centre, Italy
Christoph Grimm, University of Kaiserslautern, Austria
Michael Grottke, University of Erlangen-Nuernberg, Germany
Vic Grout, Glyndwr University, UK
Ensar Gul, Marmara University, Turkey
Richard Gunstone, Bournemouth University, UK
Zhensheng Guo, Siemens AG, Germany
Ismail Hababeh, German Jordanian University, Jordan
Shahliza Abd Halim, Lecturer in Universiti Teknologi Malaysia, Malaysia
Herman Hartmann, University of Groningen, The Netherlands
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Peizhao Hu, NICTA, Australia
Chih-Cheng Hung, Southern Polytechnic State University, USA
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Noraini Ibrahim, Universiti Teknologi Malaysia, Malaysia
Anca Daniela Ionita, University "POLITEHNICA" of Bucharest, Romania
Chris Ireland, Open University, UK
Kyoko Iwasawa, Takushoku University - Tokyo, Japan
Mehrshid Javanbakht, Azad University - Tehran, Iran
Wassim Jaziri, ISIM Sfax, Tunisia

Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA
Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraiba,Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong
Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran
Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan

Kewei Sha, Oklahoma City University, USA
Simeon Simoff, University of Western Sydney, Australia
Jacques Simonin, Institut Telecom / Telecom Bretagne, France
Cosmin Stoica Spahiu, University of Craiova, Romania
George Spanoudakis, City University London, UK
Cristian Stanciu, University Politehnica of Bucharest, Romania
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy
Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal
Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velev, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Martin Zinner, Technische Universität Dresden, Germany

International Journal on Advances in Software
Volume 13, Numbers 3 & 4, 2020

CONTENTS

Hisayoshi Ito, Graduate School of Software and Information Science, Iwate Prefectural University, Japan
Takahiro Tomisawa, Hitachi Ind. & Ctrl. Solutions Yokohama, Japan
Takashi Imabuchi, Office of Regional Collaboration, Iwate Pref. Univ., Japan

International Journal on Advances in Software, vol 13 no 3 & 4, year 2020, http://www.iariajournals.org/software/

Finding Better Matches: Improving Image Retrieval with EFM-HOG

Sugata Banerji
Mathematics and Computer Science
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045, USA
Email: banerji@lakeforest.edu

Ryan R. Zunker
Mathematics and Computer Science
Lake Forest College
555 North Sheridan Road
Lake Forest, IL 60045, USA
Email: zunkerrr@lakeforest.edu

Atreyee Sinha
Computing and Information Sciences
Edgewood College
1000 Edgewood College Drive
Madison, WI 53711, USA
Email: asinha@edgewood.edu

***Abstract*—Retrieving images from a dataset, which are similar to a query image, is an important high-level vision problem. Different tasks define similarity based on various low-level features, such as shape, color, or texture. In this article, we focus on the problem of image retrieval of similarly shaped objects, with the query being an object selected from a test image at run-time. Towards that end, we propose a novel shape representation and associated similarity measure, which exploits the dimensionality reduction and feature extraction methods of Principal Component Analysis (PCA) and Enhanced Fisher Model (EFM). We demonstrate the effectiveness of this representation on three shape-matching problems using multiple large-scale image datasets and also compare its retrieval performance with the Histograms of Oriented Gradients (HOG). Furthermore, to test the performance of our presented descriptor on the non-trivial task of image-based geo-localization, we create a large-scale image dataset and conduct extensive experiments on it. Finally, we establish that our proposed EFM-HOG not only works well on this new dataset, but also significantly improves upon the conventional HOG results.**

***Keywords*–Histogram of Oriented Gradients; Enhanced Fisher Model; Content-Based Image Retrieval; Shape Matching; EFM-HOG.**

I. INTRODUCTION

With the enormous popularity of digital devices equipped with cameras, along with the wide access to high speed Internet and cloud storage, several applications based on image search and retrieval have emerged. Such applications include augmented reality, geo-localization, security and defense, educational uses, to name a few. Billions of images are uploaded and shared over social media and web sharing platforms everyday, giving rise to a greater need for systems that can retrieve images similar to a query image from a dataset. Traditional approaches of content-based image retrieval are based upon low level cues such as shape, color and texture features. In this extended work, we address three image retrieval problems, which are all based on shape similarity. Specifically, we select a window surrounding an object of interest from a query image and want to be able to retrieve other images in the dataset, which have similarly shaped objects. Towards that end, we investigate and propose a novel EFM-HOG representation and retrieval technique [1] that is based on shape features, dimensionality reduction, and discriminant analysis. It is also robust to the slight changes in the window object selection.

The Histograms of Oriented Gradients (HOG) feature is very popular for shape matching. Simple HOG matching, however, poses significant challenges in effective image retrieval due to the fact that the apparent shape of the query object may change considerably between images due to differences in lighting, camera parameters, viewing angle, scale and occlusion. In this work, we introduce a novel technique of improving the HOG features to reduce the number of false matches.

Shape matching can be used in a variety of different scenarios and we demonstrate the effectiveness of the proposed method in three such scenarios here. The first is a simple object retrieval problem where the goal is to retrieve images of objects belonging to the same class as, or more broadly speaking, similar in shape to the query image. We use the publicly available PASCAL VOC 2012 [2] image dataset for this task. The second task that we use our method for is building image retrieval and landmark recognition. Here it is important to fetch not other similarly-shaped buildings but other pictures of the exact same building. It is also important to fetch multiple instances of the building in top search results so that the building can be identified without doubt using a k-nearest neighbors method. We use another large public database, the Oxford Buildings dataset [3] for this task.

The third problem that we try to solve using the proposed method is that of image-based geo-localization at the scale of a city. To make this problem more challenging than landmark recognition, we created a new dataset [4] based on Google StreetView images of the city of Lake Forest, Illinois, USA. This dataset has $10,000$ images and it is more challenging than a similar dataset built from a big city due to an extremely high amount of vegetation cover. Here, we had to address the problem of isolating the buildings in the images and discard most of the trees and other things. We had to design a form of coarse semantic segmentation as a preprocessing step on this dataset for this purpose.

The rest of this paper is organized as follows. Section II presents a short survey of other methods employed in shape-based image retrieval, with a brief mention of other researchers working on related problems. Section III and its subsections outline in detail the method proposed in this paper. The problems addressed and the datasets used are described in detail in Section IV. The experiments performed and results obtained are detailed in Section V. Finally, we list our conclusions and directions for future research in Section VI.

II. RELATED WORK

The HOG feature vector [5], proposed nearly two decades ago for pedestrian detection problems, has been very popular

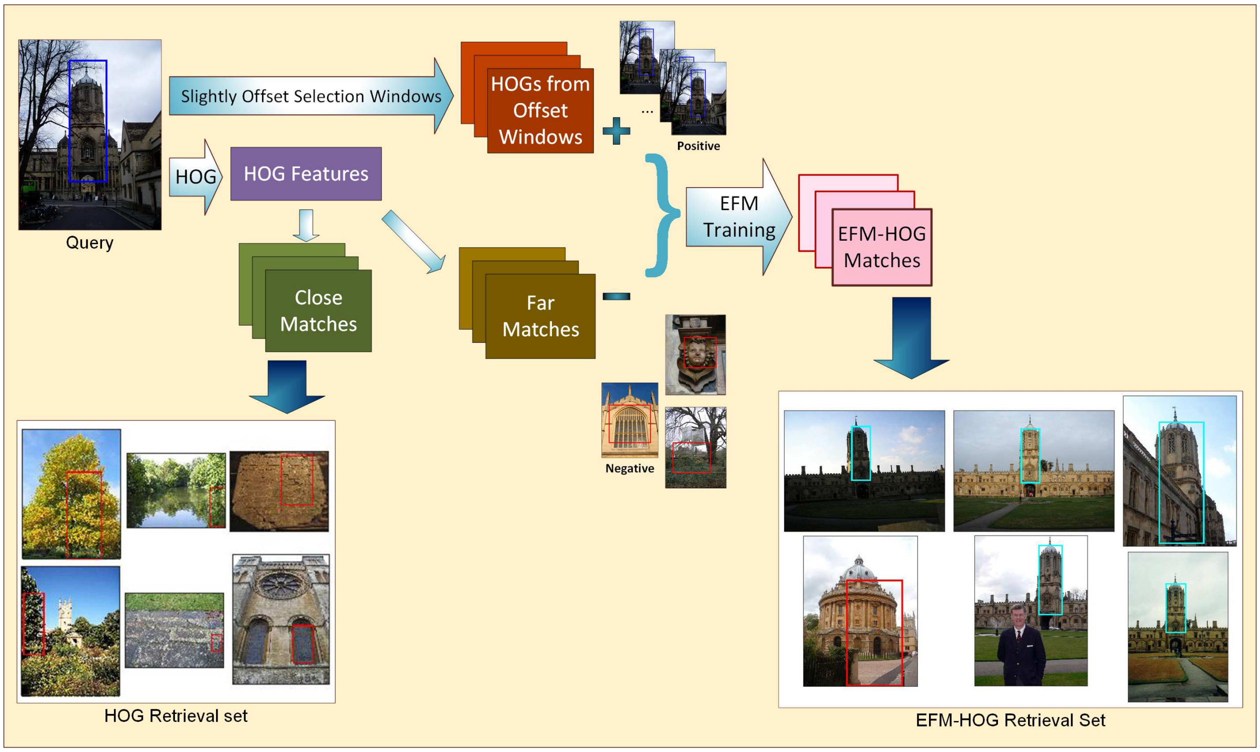


Figure 1: The process of generating the retrieval set using the proposed EFM-HOG match technique.

among Computer Vision researchers for representing shape. It has successfully been combined with other techniques [6] and fused with other descriptors [7] for classifying both indoor and outdoor scene images. HOG has also given rise to other extremely successful object detection techniques, such as Deformable Part Models (DPM) [8]. More complicated descriptors [9] [10] [11] have been used for image retrieval with reasonable success. However, such methods are time consuming and more processor-intensive as compared to simple HOG matching. In recent years, handcrafted features have declined in popularity due to the success of deep neural networks in object recognition [12] [13], but such methods are not without their drawbacks. Deep neural networks require a lot of processor time and run better on specialized hardware. They also require far greater number of training images that are available in a small or medium-sized dataset to avoid overfitting. For these reasons, enhancing simple handcrafted features like HOG can be effective for solving small-scale retrieval problems more effectively than methods of greater complexity.

The use of HOG for shape matching is fraught with challenges as mentioned above in Section I. The difficulties of using HOG for shape-based image retrieval are particularly evident for content generated by users in the wild, but are also applicable to more controlled images such as Google StreetView [14] images due to seasonal differences in vegetation and lighting. In effect, every query image is an exemplar of its own class and a retrieval system must be trained to treat it that way. In [15], this idea is handled using a Support Vector Machine (SVM) [16]. Instead of an SVM, here we introduce the novel idea of enhancing the HOG features by the Enhanced Fisher Model (EFM) process [17] because it produces a low-dimensional representation, which is important from the computational aspect. Principal Component Analysis (PCA) has been widely used to perform dimensionality reduction for image indexing and retrieval [17]. The EFM feature extraction method has achieved good success rates for the task of image classification and retrieval [7]. In the proposed method, which is represented schematically in Figure 1, we show this method to be effective in isolating the query object from the background clutter as well.

The geo-localization problem has been addressed by many researchers with varying degrees of success since the end of the last decade. The works range in scale between [18] where the authors explore the distinguishable architectural features of cities to [19] [20] where the scale is global Earth. But our work brings the problem to the scale of identifying individual buildings on Google StreetView [14] and tries to solve it. This is most similar to the work of [21], but our method uses very few (< 10) boxes per image using our proposed EFM-HOG representation.

We design a coarse semantic segmentation algorithm and use it as a preprocessing step on the dataset that we built from Google StreetView images for testing our technique. Semantic segmentation of outdoor scene images into a small number of semantic categories has been addressed successfully by [22]. While they use color histograms in the RGB and HSV color space, texture, shape, perspective and SIFT features at the superpixel level to assign pixel-level semantic labels, this was not necessary in our case. HOG features are extracted from rectangular windows and it was sufficient to achieve enough coarse semantic segmentation to draw a rectangular

Figure 2: Auto-generation of offset windows to be used as positive training samples during querying. The window dimensions and offsets shown are only representative.

bounding box around the houses, and hence we used fewer features. Local Binary Patterns (LBP) [23] is known to provide good features for not only texture but also object and scene classification [24] [25] and so LBP was chosen as the primary feature to represent patches. We also use HSV color histogram and HOG feature vectors and concatenate them to LBP for this purpose. While deep neural networks have proven very successful for the semantic segmentation task [26] [27] [28] [29], they require a large number of training images with labeled ground truth. Even weakly supervised methods [30] require a large number of images labeled at the bounding-box level and do not work well with non-convex regions such as vegetation. Since we do not have a sufficient number of images with labeled ground truth data, and neural networks trained on other cities were found to perform poorly on our new Lake Forest StreetView dataset, we do not use deep neural networks for this work.

III. Proposed Method

The proposed method, as outlined in Figure 1, works by matching HOG features [5] from a selected window in a query image surrounding an object of interest with the HOG features extracted from the similarly shaped objects in other images of the dataset. The following few subsections explain in detail the various steps needed for the proposed feature extraction and retrieval process.

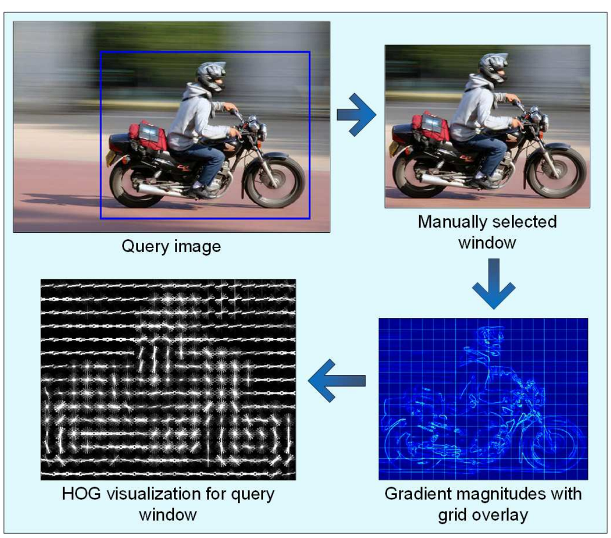


Figure 3: Formation of the HOG descriptor from a query image window.

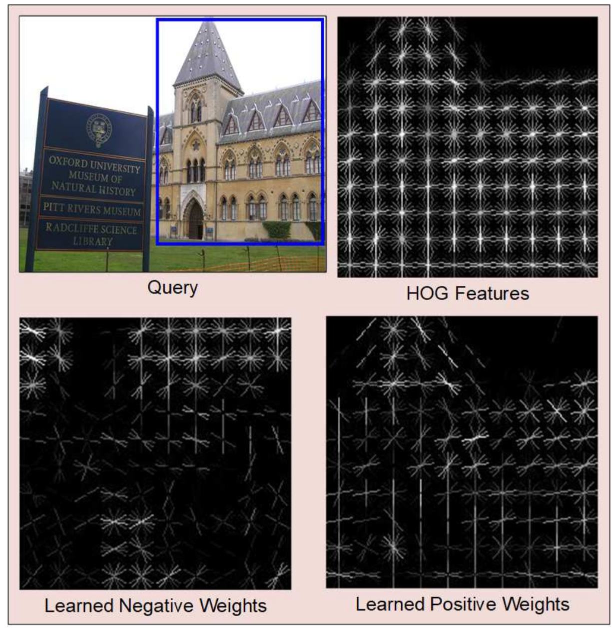
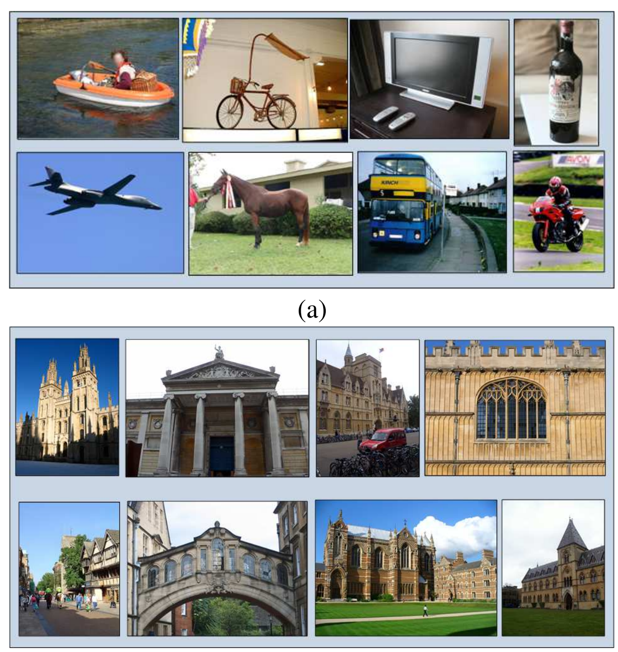


Figure 4: The positive and negative weights learned from the HOG features through the EFM discriminative feature extraction process.



(a)

(b)

Figure 5: Some sample query images from (a) the PASCAL VOC 2012 dataset, and (b) the Oxford Buildings dataset.

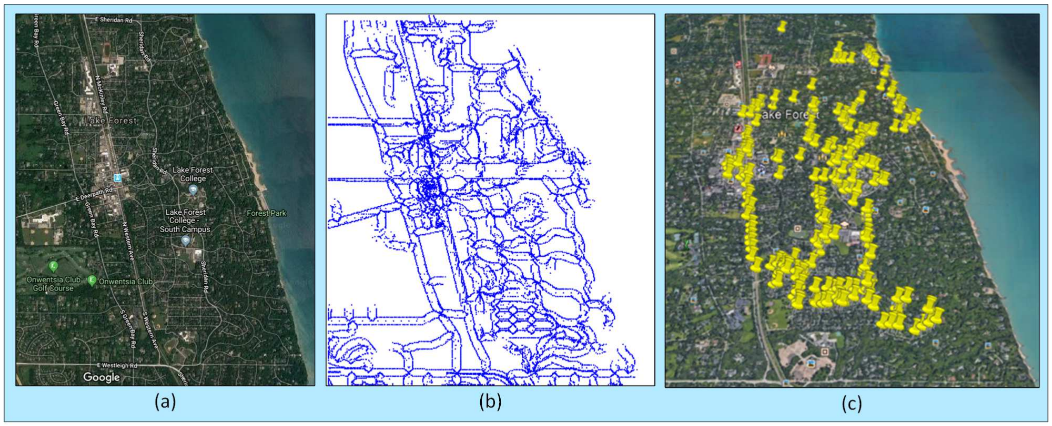


Figure 6: The locations of the images in our dataset. (a) shows the map of Lake Forest. (b) shows the distribution of the Google StreetView images collected. (c) shows the locations of our query images.

A. Window Generation

We start with generating objectness windows from each image. We use different methods on different datasets for this purpose. For the PASCAL VOC 2012 dataset [2] and the Oxford Buildings dataset [3], the method used by [31] is found to work well. This method designs an objectness measure and explicitly trains it to distinguish windows containing an object from background windows. It uses five objectness cues – namely, multi-scale saliency, color contrast, edge density, superpixels straddling, and location and size – and combines them in a Bayesian framework. We select the 25 highest-scoring windows from each image in our dataset and extract HOG features from these windows.

For the Lake Forest StreetView dataset introduced by us, the objectness method by [31] does not work well. For this dataset, we generate windows of interest by a combination of a patch-wise semantic segmentation and heuristics-based algorithm described in more detail in Section IV-C1. This method typically produces less than 10 windows per image. Whatever method we use for generating the windows, the window coordinates are pre-calculated and stored for each image file in a dataset.

While testing our system, the user generates a window on the query image manually roughly enclosing the object of interest. Then, we automatically select 10 slightly offset versions of this window. Eight of these are generated by moving the user-selected window to the right, left, up, down, up-right, up-left, down-right and down-left by 5%, respectively. Two windows are generated by expanding and contracting the user's selection by 5%, respectively. Features are now extracted from these 10 as well as the original window for further processing. This process is represented in Figure 2.

B. The HOG Descriptor

The idea of HOG rests on the observation that local features such as object appearance and shape can often be characterized well by the distribution of local intensity gradients in the image [5]. HOG features are derived from an image based on a series of normalized local histograms of image gradient orientations in a dense grid [5]. The final HOG descriptors are formed by concatenating the normalized histograms from all the blocks into a single vector.

Figure 3 demonstrates the formation of the HOG vector for a window selected from an image. We use the HOG implementation in [32] for both generating the descriptors and rendering the visualizations used in this paper.

C. Dimensionality Reduction

PCA, which is the optimal feature extraction method in the sense of the mean-square-error, derives the most expressive features for signal and image representation. Specifically, let $\mathcal{X} \in \mathbb{R}^N$ be a random vector whose covariance matrix is defined as follows [33]:

$$S = \mathcal{E}\{[\mathcal{X} - \mathcal{E}(\mathcal{X})][\mathcal{X} - \mathcal{E}(\mathcal{X})]^t\} \tag{1}$$

where $\mathcal{E}(\cdot)$ represents expectation and t the transpose operation. The covariance matrix S is factorized as follows [33]:

$$S = \Phi\Lambda\Phi^t \tag{2}$$

where $\Phi = [\phi_1\phi_2 \cdots \phi_N]$ is an orthogonal eigenvector matrix and

$$\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$$

a diagonal eigenvalue matrix with diagonal elements in decreasing order. An important application of PCA is the extraction of the most expressive features of $\mathcal{X}$. Towards that end, we define a new vector $\mathcal{Y}$: $\mathcal{Y} = P^t\mathcal{X}$, where $P = [\phi_1\phi_2 \ldots \phi_K]$, and $K < N$. The most expressive features of $\mathcal{X}$ thus define the new vector $\mathcal{Y} \in \mathbb{R}^K$, which consists of the most significant principal components.

D. EFM

The features obtained after dimensionality reduction by PCA as discussed in Section III-C are the most expressive features for representation. However, they are not the optimum features for classification. Fisher's Linear Discriminant (FLD), a popular method in pattern recognition, first applies PCA for dimensionality reduction and then discriminant analysis for feature extraction. Discriminant analysis often optimizes a criterion based on the within-class and between-class scatter matrices S_w and S_b, which are defined as follows [33]:

$$S_w = \sum_{i=1}^{L} P(\omega_i)\mathcal{E}\{(\mathcal{Y} - M_i)(\mathcal{Y} - M_i)^t|\omega_i\} \tag{3}$$

$$S_b = \sum_{i=1}^{L} P(\omega_i)(M_i - M)(M_i - M)^t \tag{4}$$

where $P(\omega_i)$ is *a priori* probability, ω_i represents the classes, and M_i and M are the means of the classes and the grand mean, respectively. One discriminant analysis criterion is J_1: $J_1 = tr(S_w^{-1}S_b)$, and J_1 is maximized when Ψ contains the eigenvectors of the matrix $S_w^{-1}S_b$ [33]:

$$S_w^{-1}S_b\Psi = \Psi\Delta \tag{5}$$

where Ψ, Δ are the eigenvector and eigenvalue matrices of $S_w^{-1}S_b$, respectively. The discriminating features are defined by projecting the pattern vector $\mathcal{Y}$ onto the eigenvectors of Ψ:

$$\mathcal{Z} = \Psi^t\mathcal{Y} \tag{6}$$

$\mathcal{Z}$ thus contains the discriminating features for image classification.

The FLD method, however, often leads to overfitting when implemented in an inappropriate PCA space. To improve the generalization performance of the FLD method, a proper balance between two criteria should be maintained: the energy criterion for adequate image representation and the magnitude criterion for eliminating the small-valued trailing eigenvalues of the within-class scatter matrix. The EFM improves the generalization capability of the FLD method by decomposing the FLD procedure into a simultaneous diagonalization of the within-class and between-class scatter matrices [17]. The simultaneous diagonalization demonstrates that during whitening, the eigenvalues of the within-class scatter matrix appear in the denominator. As shown by [17], the small eigenvalues tend to encode noise, and they cause the whitening step to fit for misleading variations, leading to poor generalization performance. To enhance performance, the EFM method preserves a proper balance between the need that the selected eigenvalues account for most of the spectral energy of the raw data (for representational adequacy), and the requirement that the eigenvalues of the within-class scatter matrix (in the reduced PCA space) are not too small (for better generalization performance). For this work, the number of eigenvalues was empirically chosen.

TABLE I: The number of images in each class of the PASCAL VOC 2012 dataset

Object Category	Number of Images
aeroplane	670
bicycle	552
bird	765
boat	508
bottle	706
bus	421
car	1161
cat	1080
chair	1119
cow	303
dining table	538
dog	1286
horse	482
motorbike	526
person	4087
potted plant	527
sheep	325
sofa	507
train	544
TV/monitor	575

TABLE II: The number of images containing each landmark in the Oxford Buildings dataset

Landmark	Good	OK	Junk
All Souls Oxford	24	54	33
Ashmolean Oxford	12	13	6
Balliol Oxford	5	7	6
Bodleian Oxford	13	11	6
Christ Church Oxford	51	27	55
Cornmarket Oxford	5	4	4
Hertford Oxford	35	19	7
Keble Oxford	6	1	4
Magdalen Oxford	13	41	49
Pitt Rivers Oxford	3	3	2
Radcliffe Camera Oxford	105	116	127

E. Training

The EFM feature extraction method uses positive and negative training samples to find the most discriminative features. In our setting, there is only one query image to be used as a positive sample. This is similar to the Exemplar-SVM training scenario used by [15], but to make the training more robust to selection error by the user and to prevent overfitting, we use 11 windows instead of just the one selected by the user as described in Section III-A.

We rank all region of interest (ROI) windows from all images in the dataset in terms of Euclidean distance in the HOG space from the original query window. For the negative training samples, we use 110 windows that are ranked low, i.e., are very distant in the HOG space. Experimentally, we found that the windows that are ranked last (i.e., farthest from the query) in the dataset are not very good candidates for negative training samples, since they are often outlier windows that contain large blank areas like the sky. Instead, windows that have a rank of $1,000$ to $5,000$ when sorted by increasing HOG distance were seen to perform well. We also tried training the system with different numbers of negative samples and found a number close to 100 performs the best. These windows are mostly background regions like ground and vegetation. The positive and negative weights for the HOG features learned by this method can be seen in Figure 4.

For an n-class problem, the EFM process for discriminatory feature extraction reduces the dimensionality of any vector to $n - 1$. Since our problem is a two-class problem, EFM produces one feature per window. We compute the score of each window by finding the absolute value of the difference between the window EFM feature and the average positive training set EFM feature. Ranking the images by their best-scoring windows gives us the retrieval set.

IV. Shape Retrieval Tasks and Datasets

To prove the effectiveness of our proposed EFM-HOG descriptor and the associated distance measure, we apply it

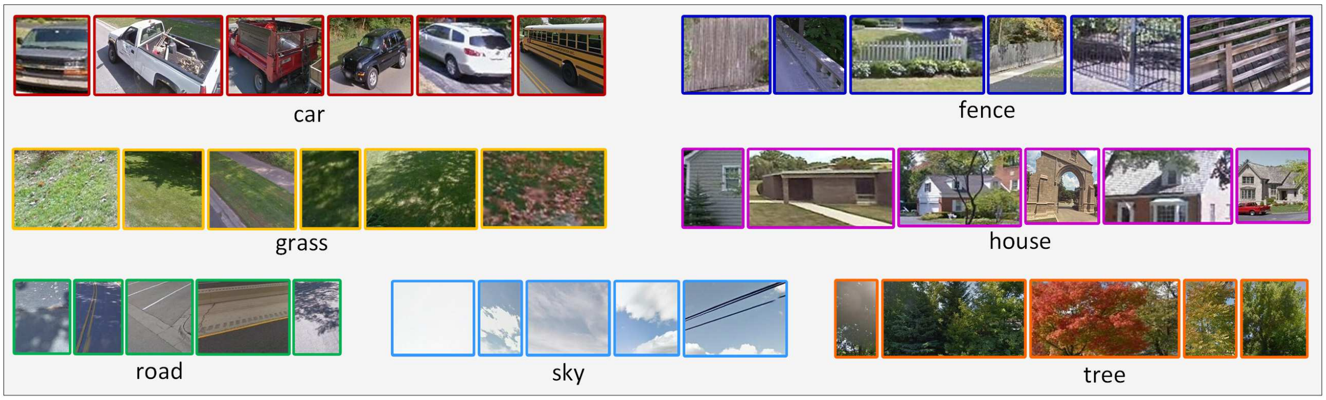


Figure 7: Manually selected training patches used to train the seven SVM classifiers for coarse semantic segmentation.

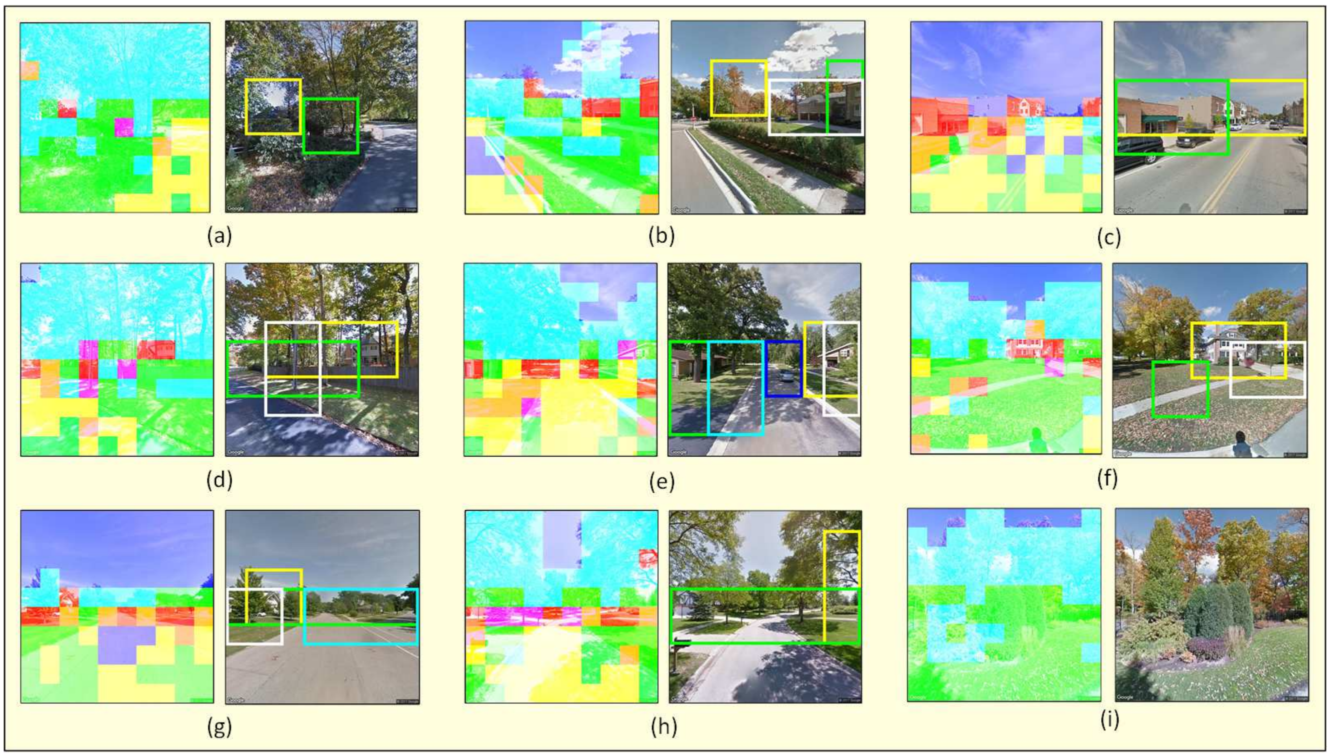


Figure 8: Some examples of ROI selection from our reference dataset. The left-side images in (a) through (i) show different semantic categories by using different colors. The red patches indicate the house category and magenta indicates fence. The right-side images show the output of our ROI-window generation algorithm. Colors of the rectangles in the right-side images have no significance. Note that in (i), no buildings are found, and so no windows are generated.

to three distinct problems. For each of these tasks, we use a different dataset with properties suitable for the problem being addressed. In this section, we will give a brief description of the different problems addressed and datasets used for our experiments, and then we will discuss the performance of our novel EFM-HOG matching algorithm on these datasets in the next section.

A. *Object Search and Retrieval*

The first problem that we address is that of object search and retrieval. In this problem, the user selects a bounding box around an object in a query image and we attempt to retrieve similar objects from the dataset. For this task, we use PASCAL VOC 2012 dataset [2]. We only use the training/validation data from this dataset to test our retrieval algorithm. This data consists of $11,540$ images from 20 classes (many images have multiple classes present). The classes in this image dataset are aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train and TV/monitor. The classes and the number of images in them are shown in Table I. Figure 5(a) shows some images from this dataset. We create five randomly selected

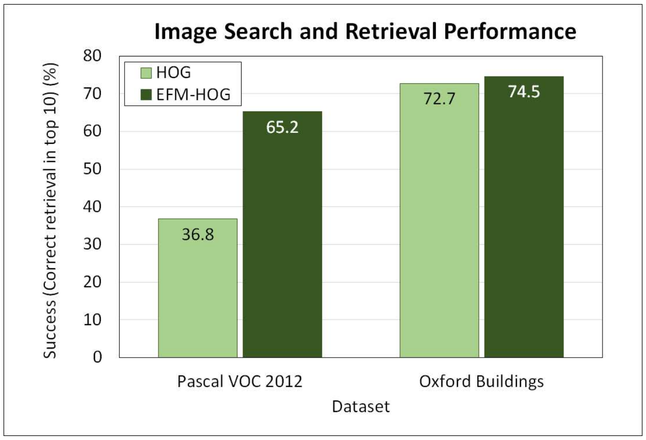


Figure 9: Mean retrieval accuracy (measured by the presence of a relevant image in the top 10 retrieved images).

100-image test sets from the dataset and perform a five-fold cross-validation. A successful retrieval experiment is one where the program retrieves at least one relevant image (an image containing the query object) within the top 10 results. The performance of our descriptor on this dataset is discussed in Section V.

B. Landmark Recognition in the Wild

The second problem that we address is that of landmark recognition in the wild. This is more challenging than the object search and retrieval problem because the images here are mostly outdoor images, and buildings are not always as easily distinguishable from their surroundings as object images are. The dataset that we use for this problem is the Oxford Buildings dataset [3], which consists of 5, 062 images of 11 different Oxford landmarks and distractors collected from Flickr [34]. 55 images from this dataset were used as queries for testing our retrieval system. Flickr images are completely user-generated, which means there is a great variation in camera type, camera angle, scale and lighting conditions. This makes this dataset very difficult for image retrieval in general and landmark-recognition in particular. Figure 5(b) shows some of our query images from this dataset. For each query, the images that contain the query landmark are further classified into *good*, *OK* and *junk* categories, with progressively poorer views of the query landmark. Table II shows the landmark-wise distribution of *good*, *OK* and *junk* images in this dataset.

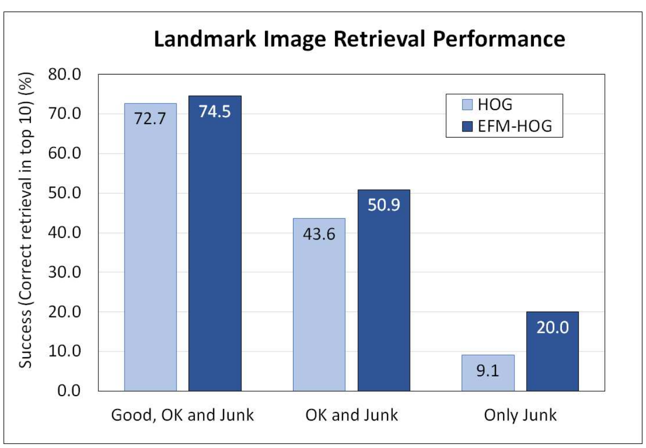


Figure 10: Landmark image-retrieval accuracy (measured by the presence of a relevant image in the top 10 retrieved images) on the Oxford Buildings dataset. The three sets of values show the success rate of HOG and EFM-HOG while varying the quality of available matches in the reference set.

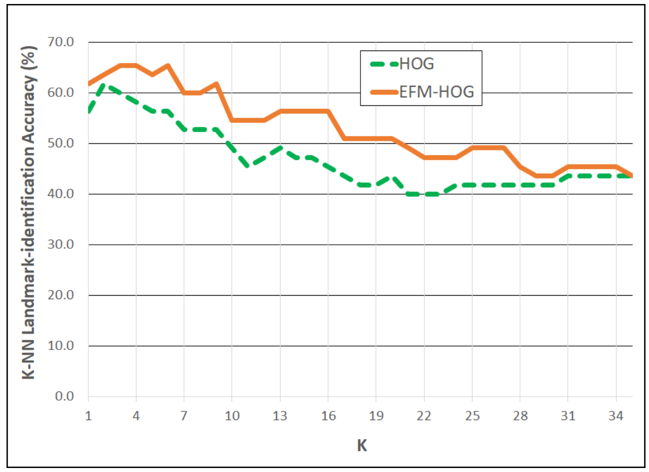


Figure 11: The mean landmark-recognition performance on the Oxford Buildings dataset by using the k-nearest neighbors method with varying k.

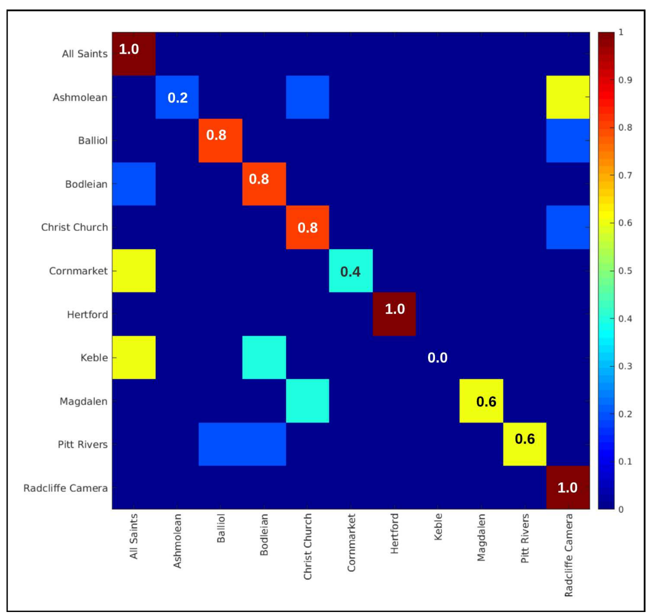


Figure 12: The confusion matrix for the landmark-recognition performance of the EFM-HOG descriptor on the Oxford Buildings dataset by using the k-nearest neighbors method with k=3.

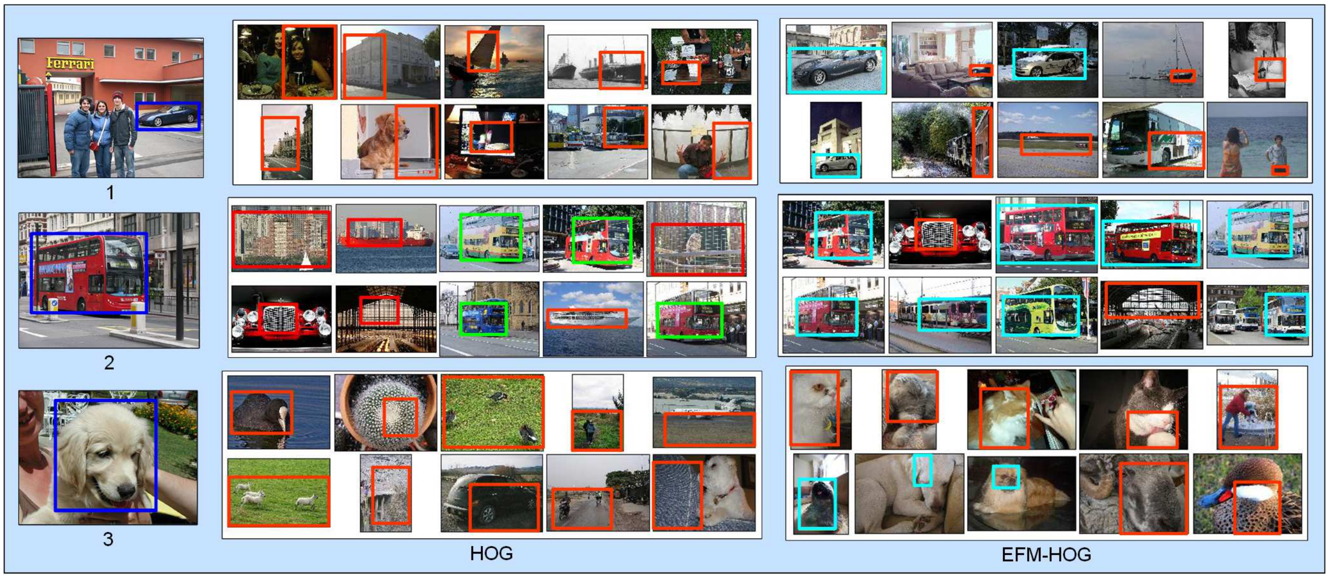


Figure 13: Comparison of image retrieval results for HOG and the proposed EFM-HOG on the PASCAL VOC 2012 dataset.

C. Image-based Geo-Localization

Finally, to test our proposed algorithm for image-based geo-localization of buildings within an entire city using StreetView images, we built a new image dataset [4], which we call the Lake Forest StreetView dataset, from the city of Lake Forest, Illinois. We collected two sets of images for this purpose: one for query images, and one for reference images. We acquired our own images for the query dataset by walking around the city and taking photos of buildings with smartphone cameras. This dataset has 308 images. For the reference dataset, we downloaded Google StreetView [14] images from around the city. We downloaded eight overlapping StreetView images from points eight feet apart along every road in Lake Forest. This process downloaded 126,000 images. From our 308 query images, we selected 128 images spread over the whole city of Lake Forest that contained buildings that were also visible in at least one of the reference images. To do this, we wrote a program that uses the GPS tags on each query image to retrieve the geographically nearest 100 images from the reference dataset. We then visually inspected this retrieved set to determine if the query image building was visible in any of them. Finally, we combined these retrieved sets together, eliminated duplicates, and added a few thousand random distractor StreetView images to bring the total up to 10,000 images. This was our final reference set for the experiments. Figure 6(a) shows the Google Maps view of Lake Forest. Figure 6(b) shows the distribution of our reference set, which is composed of downloaded Google StreetView images. Each blue point in this image represents the location of a Google StreetView photo. It can be seen that our image dataset follows the streets and there are large areas without any images in between, which are private estates and parks. Finally, Figure 6(c) shows the distribution of our query images using yellow markers. These markers were generated directly using the GPS tags of the query images, which were taken using smartphone cameras.

We ran retrieval experiments on this set using each of the 128 query images. We manually drew rectangles around buildings in each of the query images, which were then used to extract the EFM-HOG features for matching. The process of selecting multiple windows that are slightly offset from the original reduces the impact of slight variations between manually drawn rectangles in two experiments, but still the manually drawn rectangle boundaries were saved to preserve repeatability between experiments. Degree of success or failure of a retrieval was measured by the mean geographical distance of retrieved images from the query, and also by the presence or absence of the query building in the retrieval set.

1) ROI Selection for Geo-localization Problem: HOG matching starts with selecting a bounding box around the ROI, which in this case would be the buildings. The task of selecting the buildings in Lake Forest, however, is non-trivial due to a characteristic of the city itself. The city of Lake Forest has a very large number of trees and most of the houses are far from the road in the middle of large estates. The Google StreetView images are shot with a wide-angle camera mounted on a moving car. The combination of a wide-angle lens and the large distance from the road causes the houses to appear very small in the images, and the vegetation or parts of the road closer to the camera appear much larger. In majority of the reference images, the buildings occupy only a small portion of the image, the rest being filled with vegetation, sky or portions of the road. Hence, selecting an ROI containing the building becomes an important preprocessing step before features can be extracted.

The object detection program used on the other two datasets did not work well on this dataset, and we needed some coarse form of semantic segmentation to separate the houses from the vegetation, road and other objects. We did not use a deep neural network for this purpose because of two reasons. First, we did not need pixel-level separation of categories since HOG features are extracted from rectangular windows anyway. Second, we did not have labeled segmentation ground truth training images and networks trained on images from other

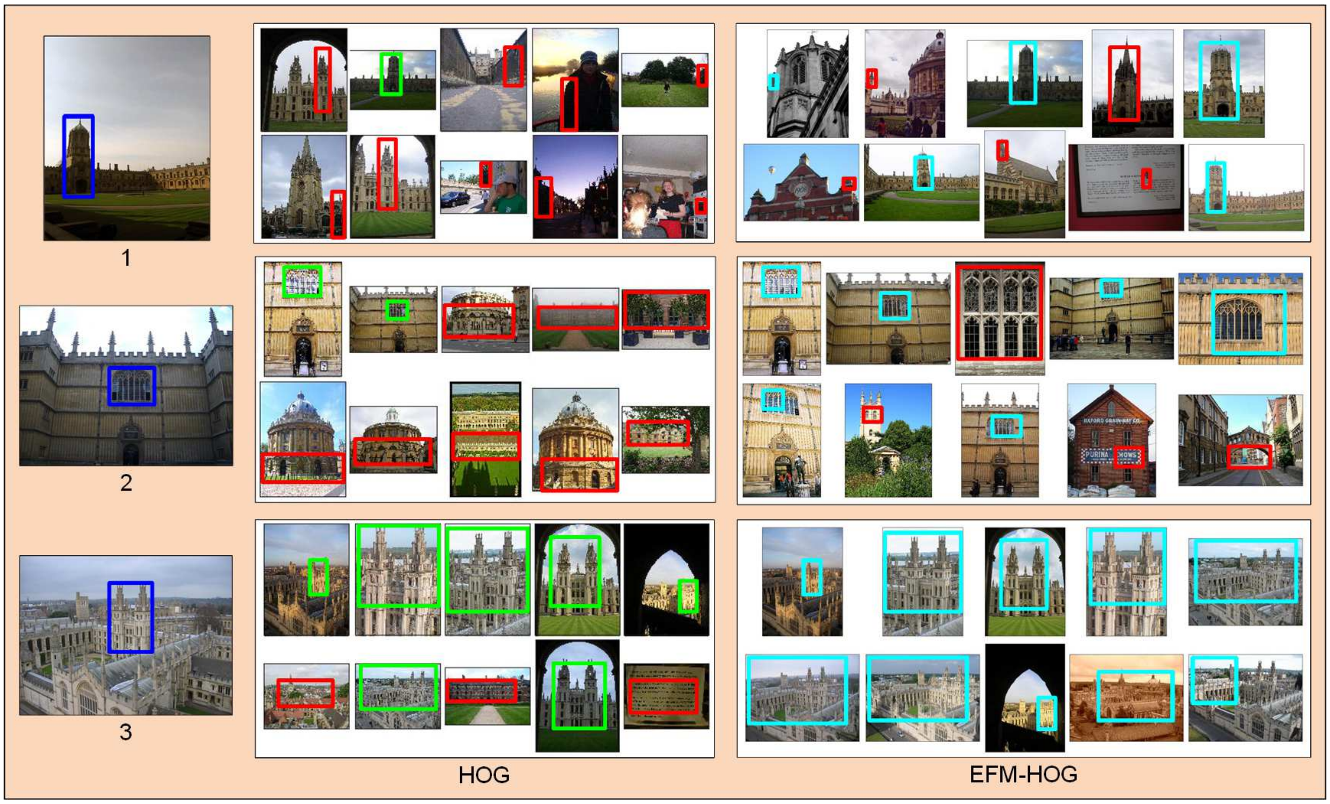


Figure 14: Comparison of landmark image retrieval results for HOG and the proposed EFM-HOG on the Oxford Buildings dataset.

cities did not generalize well to Lake Forest. So, we built our own semantic segmentation technique for this step.

On visual inspection of the images, we decided there were seven major semantic classes, namely sky, grass, tree, road, house, fence and vehicles. We manually selected rectangular patches from each of these classes and extracted three sets of features from each patch. These features are color histogram in the HSV color space, HOG and LBP. These three sets of features are concatenated to get our feature vector to train the classifiers for coarse semantic segmentation. For this task, we trained an SVM [16] classifier for each class.

2) The Linear SVM Classifier: The SVM is a particular realization of statistical learning theory. The approach described by SVM, known as structural risk minimization, minimizes the risk functional in terms of both the empirical risk and the confidence interval [16]. The SVM implementation used for our experiments is the one that is distributed with the VlFeat package [32]. We use the one-vs-all method to train an SVM for each semantic category. The parameters of the SVM are tuned empirically using only the training data, and the parameters that yield the best average precision on the training data are used for classification of the test data.

We created the training data for our SVMs in the form of rectangular windows selected manually from the reference images. 100 training patches were used per class. Some of these patches are shown in Figure 7. We divide each reference image into 100 uniformly sized patches over a 10×10 regular grid and pass each patch through all seven classifiers to assign one final label to each patch. Finally, we draw minimal bounding boxes around the house and fence category patches (if any) with some padding around them, and extract HOG features from them. This process is shown in the different parts of Figure 8. The bounding boxes in Figure 8(a) look like they are enclosing vegetation, but there are underlying house and fence outlines visible here through the trees. Bounding boxes are not perfect, and sometimes they either enclose objects other than houses, like the blue box in Figure 8(e), or the green box in Figure 8(f), or are too large, like the green box in Figure 8(h). False positives, however, are less of a problem to our technique than false negatives, and extra bounding boxes are better than missing houses. Also note that in Figure 8(i), there are no buildings, and no bounding boxes are generated, which is a strong indication that our segmentation algorithm is successful in cutting down on the number of undesirable windows in a large number of cases. If we had used the objectness code that we ran on the other two datasets, we would have got 25 windows from this image as well. The different colors of the bounding boxes in the right-side images in Figure 8 have no special significance. The various colors have been used to differentiate between the rectangles.

V. Experiments and Results

In this section, we describe in detail the experiments that we performed on our three datasets and the results that we obtained in each of the three tasks that we attempted. For each of the tasks, we used a different dataset and compared the results of our EFM-HOG descriptor with that obtained

Figure 15: Comparison of image-based geo-localization results for HOG and the proposed EFM-HOG on the Lake Forest StreetView dataset.

by conventional HOG features. We also provide samples of query images along with top matching images retrieved by both algorithms from all three datasets for a qualitative comparison of the results.

A. The Object Search and Retrieval Task

The proposed image representation is tested on three different tasks, the first of which is object search and retrieval. Here, an image is used as a query to retrieve similar scenes from the dataset. For this, the user selects a rectangular ROI from the query image, and HOG features from this rectangular window are matched with the 25 highest scoring objectness windows from each image in the database, both in the raw HOG space and in the EFM-HOG space after the proposed training and feature extraction procedure. The closest matches based on Euclidean distance are retrieved in order of their distance from the query window. Finding an instance of the query class object in the top 10 retrieved images is considered a success. Figure 9 compares the retrieval success rates of the HOG descriptor and the proposed EFM-HOG representation on the PASCAL VOC 2012 dataset. For this dataset, the retrieval experiment is performed on five random splits and the average success rate is found to be 65.2% for EFM-HOG as compared to 36.8% for HOG. We also find that the conventional HOG performs quite well for clearly segmented objects, such as airplanes in the sky, but the EFM-HOG performs much better for images of objects with a cluttered background.

We also experimented on the Oxford Buildings dataset from the retrieval point of view. Figure 9 also compares the retrieval success rates of the HOG descriptor and the proposed EFM-HOG representation on this dataset alongside the PASCAL VOC 2012 performance. Specifically, in 41 cases out of 55 queries in the Oxford buildings dataset (74.5% cases), the query landmark is retrieved within top 10 images by the proposed method, as opposed to 40 by HOG (72.7% cases). This is actually a very small difference, but this can be explained by the nature of this dataset. For all landmark query images in this dataset, there are at least some *Good* images in the dataset that show clear views of the landmarks with no occlusions. HOG is actually pretty effective at retrieving these images. To better demonstrate the effectiveness of the proposed method, we repeat this experiment with just the *OK* and *Junk* images, and then just the *Junk* images for each query. In these experiments, we find that the HOG method retrieves a relevant image in the top 10 much less frequently than the EFM-HOG method. With just the *Junk* files, EFM-HOG performs more than twice as well as HOG. These results are shown in Figure 10.

B. The Landmark-Recognition Task

The second experiment that we performed with the new EFM-HOG descriptor on the Oxford Buildings dataset was a landmark-recognition task where the system tries to label each query image with its correct landmark label. Some images

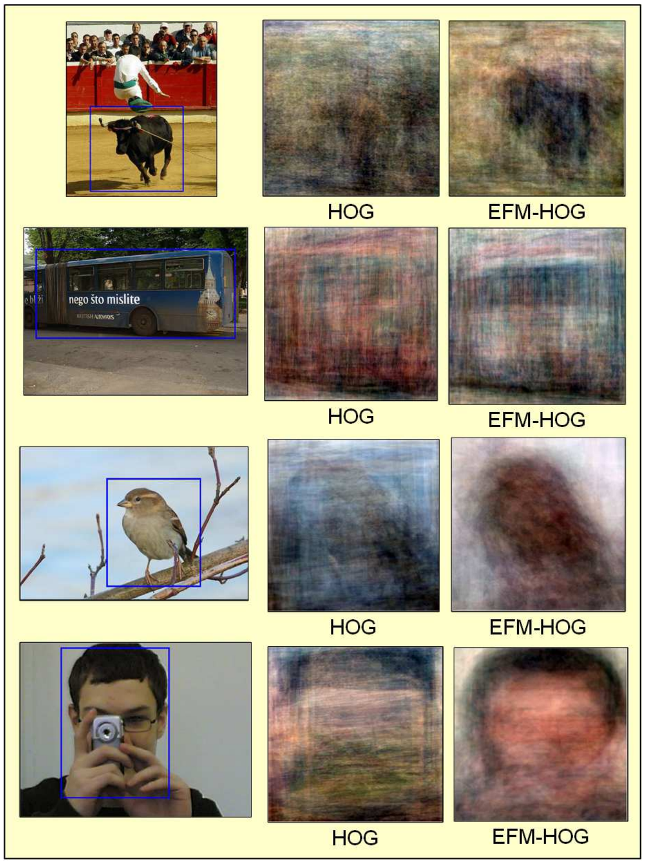


Figure 16: The means of the top 100 retrieved windows for HOG and EFM-HOG for 4 query images from the PASCAL VOC 2012 dataset.

in this dataset belong to one of the 11 landmarks listed in Table II, the others belong to none of the classes and are used as distractors. We did this task by retrieving relevant images in a manner similar to the retrieval task, and then performing the k-nearest neighbors (k-NN) classification on the top k results. The same experiments are repeated for the conventional HOG descriptor as well. As can be seen from Figure 11, the proposed EFM-HOG outperforms HOG all values of k between 1 and 35. The highest EFM-HOG landmark-recognition performance of 65.5% is achieved at k=3. A further breakdown of the landmark-recognition performance of the EFM-HOG descriptor is seen in Figure 12. Here, the rows represent real landmark labels of the queries and the columns represent predicted labels. The results are averaged over the 5 query images for each landmark and the k-NN classifier has been used with k=3.

Some HOG and EFM-HOG retrieval results on the PASCAL VOC 2012 dataset are shown in Figure 13. In this figure, the query images are shown on the left of each row with blue bounding boxes followed by the two retrieval sets obtained by using HOG and EFM-HOG. A red bounding box in a retrieved image indicates that the retrieved image is not from the same class from the query image (shown on the left). Correct matches for HOG are shown with green bounding boxes and correct matches for EFM-HOG are shown with cyan bounding boxes.

Some comparative retrieval results between HOG and EFM-HOG on the Oxford Buildings dataset are shown in Figure 14. In this figure too, the query images are shown on the left of each row with blue bounding boxes. The retrieval set in the middle of each row is obtained by using HOG and the retrieval set on the right is obtained by using EFM-HOG. A red bounding box in a retrieved image indicates that the retrieved image has a different landmark building label from the query image (shown on the left). Correct label matches for HOG are shown with green bounding boxes and correct label matches for EFM-HOG are shown with cyan bounding boxes.

C. The Geo-localization Task

We ran two sets of experiments on our Lake Forest StreetView dataset for this task. The first set does the retrieval with traditional HOG and the second set uses the proposed EFM-HOG matching. The improvement in retrieved result sets achieved by the proposed EFM-HOG technique can be seen by comparing the results shown in Figure 15. In this figure, the retrieved images have their geographical distance from the query written above them. Green text signifies a retrieved image closer than 0.1 miles, and in all the examples here, an exact match. As can be seen in all the examples, HOG fails to find even a single match for the building in the query image in the top 10 retrieval results while EFM-HOG finds one in all three.

Our EFM-HOG match program retrieved (within the top 20 results) at least one image that was closer than 100 yards (0.0568 miles) of our query in 40 out of the 128 queries that we used. In 17 of these images the exact building was found and matched. Three such query images and the top 10 retrieved images along with their geographic distances are shown in Figure 15. In the result images, a geographic distance written in green indicates an actual match. HOG is unable to retrieve a match in the top ten results in any of the three queries shown while EFM-HOG fetches one result among the top ten in all three.

D. Qualitative Analysis of Retrieved Image Windows

We also manually inspected our retrieved image windows and ran some experiments to do a qualitative analysis of the results. Figure 16 shows an interesting aspect of our retrieval technique. Here, we show the image means of the first 100 windows retrieved by both HOG and EFM-HOG on the PASCAL VOC 2012 dataset. The figure shows that the EFM-HOG means contain clearer shapes, which indicates that the EFM-HOG retrieves more similar shapes than HOG, even when the results are irrelevant to the query.

A few successfully geo-localized buildings from our retrieval experiments on the Lake Forest StreetView dataset are shown in Figure 17. In each of the image pairs shown in this figure, the left image with a red bounding box shows the query taken with a smartphone camera, and the right one with a cyan bounding box shows a retrieved image from the Lake Forest StreetView dataset. In one of these images, the match is successful even with only a small section of the fence visible in the query, which shows the technique is quite robust. The rectangles around the buildings in the retrieved images themselves were generated by our coarse semantic

Figure 17: Successfully geo-located query images along with the retrieved Google StreetView images that are exact matches for the query.

segmentation algorithm, which is also a qualitative measure of the success of this algorithm.

VI. Conclusion and Future Work

We have presented in this paper a new image descriptor based on HOG and discriminant analysis that uses a novel approach to fetch scenes with similar shaped objects. We have conducted experiments using over $5,000$ images from the Oxford Buildings dataset and over $11,500$ images from the PASCAL VOC 2012 dataset and concluded the following: (i) HOG features are not always sufficiently discriminative to perform meaningful retrieval, (ii) the discriminative nature of HOG features can be improved with the EFM for feature extraction and dimensionality reduction, and (iii) HOG features perform well for clearly isolated objects with little background clutter, but the EFM-HOG performs better for real-world images with cluttered backgrounds.

We furthermore demonstrated the effectiveness of our proposed EFM-HOG descriptor for geo-localization on a $10,000$-image Lake Forest StreetView dataset that we built from scratch. We also developed a coarse semantic segmentation strategy to automatically isolate buildings and draw bounding boxes around them as a preprocessing step before the HOG feature extraction. Finally, we compare the proposed EFM-HOG representation and the traditional HOG representation to demonstrate that our method is superior for retrieval. We intend to use this method with other image retrieval tasks in the future, so that a more thorough understanding of its strengths and weaknesses can be achieved.

It is evident that the successful geo-localization in the Lake Forest StreetView dataset depends heavily on the quality of the bounding boxes generated by our coarse semantic segmentation algorithm, and improving that algorithm will significantly improve the results. In future, we plan to develop a more robust strategy for semantic segmentation. Superpixel-based and deep neural network-based semantic segmentation may also be used if we can get a sufficient number of labeled images. We also plan to extend our dataset to cover other cities.

Acknowledgment

The authors would like to thank Professor Jana Košecká at the Department of Computer Science, George Mason University, Fairfax, Virginia for some valuable input on the EFM-HOG method and the experiments conducted.

The authors would also like to thank the Richter Scholars Program at Lake Forest College for partially supporting the research presented in this paper.

References

[1] S. Banerji and A. Sinha, "EFM-HOG: Improving Image Retrieval in the Wild," in Proceedings of the Fourth International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2019), June 2019, pp. 6–11.

[2] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," International Journal of Computer Vision, vol. 88, no. 2, 2010, pp. 303–338.

[3] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[4] R. R. Zunker, S. Banerji, and A. Sinha, "House Hunting: Image-based Geo-Localization of Buildings within a City," in Proceedings of the Fifth International Conference on Computing and Data Engineering, 2019, pp. 100–104.

[5] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.

[6] S. Banerji, A. Sinha, and C. Liu, "Scene Image Classification: Some Novel Descriptors," in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2012, pp. 2294–2299.

[7] A. Sinha, S. Banerji, and C. Liu, "Novel Color Gabor-LBP-PHOG (GLP) Descriptors for Object and Scene Image Classification," in Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, 2012, pp. 58:1–58:8.

[8] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, 2010, pp. 1627–1645.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, November 2004, pp. 91–110.

[10] H. Bay, T. Tuytelaars, and L. J. V. Gool, "Surf: Speeded up robust features," in Proceedings of the European Conference on Computer Vision, 2006, pp. 404–417.

[11] K. E. A. Van De Sande, C. G. M. Snoek, and A. W. M. Smeulders, "Fisher and VLAD with FLAIR," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2014, pp. 2377–2384.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Proceedings of the Twenty-sixth Conference on Neural Information Processing Systems, 2012, pp. 1106–1114.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of the Third International Conference on Learning Representations, 2015.

[14] "Google StreetView," https://www.google.com/maps, accessed on Mon, December 7, 2020.

[15] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of Exemplar-SVMs for Object Detection and Beyond," in Proceedings of the International Conference on Computer Vision, 2011, pp. 89–96.

[16] V. Vapnik, The Nature of Statistical Learning Theory. Springer-Verlag, 1995.

[17] C. Liu and H. Wechsler, "Robust Coding Schemes for Indexing and Retrieval from Large Face Databases," IEEE Transactions on Image Processing, vol. 9, no. 1, 2000, pp. 132–137.

[18] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes Paris look like Paris?" Communications of the ACM, vol. 58, no. 12, November 2015, pp. 103–110.

[19] J. Hays and A. A. Efros, "IM2GPS: estimating geographic information from a single image," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[20] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, "Image sequence geolocation with human travel priors," in Proceedings of the Twelfth IEEE International Conference on Computer Vision, September 2009, pp. 253–260.

[21] R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Locality in generic instance search from one example," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 2014, pp. 2099–2106.

[22] G. Singh and J. Košecká, "Introspective Semantic Segmentation," in Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2014, pp. 714–720.

[23] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," Pattern Recognition, vol. 29, no. 1, 1996, pp. 51–59.

[24] S. Banerji, A. Sinha, and C. Liu, "New image descriptors based on color, texture, shape, and wavelets for object and scene image classification," Neurocomputing, vol. 117, no. 0, 2013, pp. 173–185.

[25] ——, "A New Bag of Words LBP (BoWL) Descriptor for Scene Image Classification," in Proceedings of The Fifteenth International Conference on Computer Analysis of Images and Patterns, 2013, pp. 490–497.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015, pp. 3431–3440.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, 2017, pp. 2481–2495.

[28] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, 2018, pp. 834–848.

[29] W. Wang, Y. Fu, Z. Pan, X. Li, and Y. Zhuang, "Real-time driving scene semantic segmentation," IEEE Access, vol. 8, 2020, pp. 36 776–36 788.

[30] G. Papandreou, L. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1742–1750.

[31] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, November 2012, pp. 2189–2202.

[32] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008, accessed on Mon, December 7, 2020.

[33] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed. Academic Press, 1990.

[34] "Flickr," http://www.flickr.com, 2004, accessed on Mon, December 7, 2020.

Requirements Traceability using SysML Diagrams and BPMN

Corina Abdelahad and Daniel Riesco
Departamento de informática
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: cabdelah@unsl.edu.ar, driesco@unsl.edu.ar

Carlos Kavka
Research and Development Department
ESTECO SPA
Trieste, Italy
e-mail: kavka@esteco.com

Abstract— **An important activity in systems development is ensuring that all system requirements are met. Model-Based Systems Engineering is a methodology that benefits the documentation of the requirements and decisions that are made during the design process. On the other hand, visualizing different perspectives that focus on different aspects of the system permits to capture all the details of the design while refining the level of detail of the models. SysML is a Systems Modeling Language, which is defined as an extension of the well-known Unified Modeling Language standard. It is based on four pillars, which give the possibility to view a system from four different perspectives, supporting requirements traceability. Requirements traceability refers to the ability to describe and follow the life of a requirement in both a forward and backward direction. This traceability has an important role in Model-Based Systems Engineering. The central aim of this paper is to present a traceability approach that supports decision-making requirements. To carry out this traceability we propose to combine SysML and Business Process Model and Notation and Decision Model and Notation. SysML is used to model some aspects of system, and processes and decision-making activities are defined in terms of BPMN and DMN standards, respectively. This proposal seeks to help engineers to improve their design and enhance traceability starting from requirements, integrating and covering the different views. Our contribution is illustrated by means of a case study.**

Keywords-SysML; BPMN; DMN; requirements traceability.

I. INTRODUCTION

Abstraction is a technique used by engineers to deal with complexity, permitting them to focus only on the information that is considered significant or relevant. To improve the design of requirements, to understand and cover their different views improving maintenance and verification activities, it is necessary to carry out requirements traceability [1]. International Council on Systems Engineering (INCOSE) [2] indicates that "requirements traceability refers to the ability to describe and follow the life of a requirement in both a forward and backward direction along the design stages". Traceability plays an important role as part of any Model-Based Systems Engineering (MBSE) methodology. MBSE is a successful methodology for the design of complex systems, which emphasizes the use of models when performing systems engineering activities [3]. These models, which can be executable or not, are used to describe the structure and the behavior of the systems.

With the evolution of systems engineering, the need for a consistent standard modeling language arose. INCOSE together with the Object Management Group (OMG) [4] defined SysML, a general-purpose modeling language based on UML, which can be used for specifying, analyzing, designing, and verifying complex systems, including hardware, software, information, personnel, procedures, and facilities [5]. SysML is based on four pillars, which give the possibility to view a system from four different perspectives: Requirements, Structure, Behavior and Parametrics, each one of them defined in terms of diagrams [6]. Requirements modeling [7] is implemented in terms of the requirement diagram, which allows for capturing, analyzing and maintaining traceability of requirements in the modeled system. Structure modeling has a block definition diagram as the main diagram, representing structural elements (blocks) with their properties, relationships, and composition. Behavior modeling has different kinds of behavior diagrams like activities diagram, state machine, and sequence diagram. Parametric modeling has a parametric diagram that can be used to identify the system constraints [5]. In SysML, requirements can be related to other requirements, as well as to other model elements via one or more relationships, making possible the traceability of requirements. Furthermore, SysML can be integrated into other tools including spreadsheets and design and simulation software, such as Matlab or Modelica [8], enabling requirements verification.

The specification of business processes also followed the same path requiring standards for its definition. In particular, the Business Process Management Initiative (BPMI) together with the OMG developed the widely used BPMN notation for modeling business processes [9]. BPMN defines an abstract representation for the specification of business processes, which can include human intervention or not. BPMN couples an expressive graphical representation with a rigorous Extensible Markup Language (XML) encoding of processes and the interactions among them, supporting not only modeling activities but also process execution by using appropriate BPMN engines. Several works in the engineering field have shown the value of using BPMN instead of UML activity diagrams. Activity Diagram can be used for business process modeling but BPMN was designed exclusively for modeling business process [9] and OMG adopted BPMN instead of the Activity Diagram (UML AD) as the core standard to create a business modeling framework [10]. BPMN has model elements that, in some cases, do not have a corresponding element in UML 2.0 AD. There are cases when components of the business processes are modeled using only one symbol in BPMN and using a group of symbols in UML AD [11]. Since many activities within a business process involve decision-making, the OMG defined recently the Notation and Decision Model and Notation (DMN) standard for the elicitation and representation of decision models, effectively separating decision logic and control flow logic in business processes [12]. DMN was designed to be usable alongside the standard BPMN. At present, many companies have adopted BPMN not only because of its popularity, but because it is strongly related to DMN. This standard is already receiving adoption in the industry, with many tools being developed to assist users in modeling, checking, and applying DMN models.

As the main contribution, this work presents an innovative approach to enhance requirements traceability in the context of MBSE, by combining SysML, BPMN and DMN. This approach can help systems engineers to improve the design of requirements, to understand and cover their different views, improving maintenance and verification activities while contributing to refine the level of detail of the models.

The rest of this paper is organized as follows: Section II introduces related work, while Section III summarizes the basic concepts used in this paper. Section IV addresses the proposed approach with a case study presented in Section V. Section VI shows the conclusion.

II. RELATED WORK

Several works in the field of software engineering are related to the concept of requirements traceability using SysML. For example, the authors in [13] show how requirements traceability for mechatronic design can be achieved using MBSE and SysML. SysML is used for linking system requirements to the system elements of the different domain models while guaranteeing the traceability. This paper presents a case study of a mechatronic system in order to show this traceability.

In [14], the authors propose a Model-Based Systems Engineering approach based on SysML. This approach enables the capture and the definition of functional requirements, validate these functional requirements through functional simulation, and verify efficiently the consistency of these functional requirements. The approach is illustrated by means of a case study of an industrial avionics system.

In [15], a solution for SysML model verification and validation in an industrial context is presented. The authors provide a method and a list of the existing challenges; besides that, they show experimental results. A case study is presented, verification rules are in Object Constraint Language (OCL), while the validation rules are in a formal text format evaluated by a script. The authors mention that the verification of these rules ensures a certain degree of traceability.

In [16], an approach to construct true model-based requirements in SysML is presented. This approach proposes that every requirement can be modeled as an input/output transformation. This proposal uses SysML behavioral and structural models and diagrams, with specific construction rules derived from Wymore's mathematical framework for MBSE and taxonomies of requirements and interfaces. The authors consider that this proposal provides several benefits, including traceability, and improved precision over the use of natural language.

In [17], the authors propose a model-based approach to automotive requirements engineering for the general development of vehicles of passengers. The SysML requirement element is extended, through stereotype, to functional and non-functional requirements. The paper validates the advantages that include classified and modeled requirements graphically, as well as their relationships that are explicitly mapped. This article presents a case study that shows the proposed extension and the performed requirements traceability.

In [18], the authors propose a model-driven requirement engineering approach for the embedded software domain. This approach is based on UML, MARTE and SysML standard notations, which are integrated in order to improve requirements specification and traceability. MARTE is used to allow domain-specific non-functional requirements to improve the software specification and SysML is combined with UML/MARTE models to support requirements management, to follow their changes. The approach is illustrated by means of a case study.

In [19], the authors propose a metamodel, which establishes the traceability links among the requirement model, the solution model and the verification and validation model for embedded system design. This approach enables traceability of requirements by considering heterogeneous languages for modeling and verifying real-time embedded systems. A case study illustrates the approach with the use of languages such as SysML, MARTE, SIMULINK, among others.

However, to the best of our knowledge, no research work about requirements traceability has considered the decision requirement, a kind of requirement that involves decision making. This requirement appears in the decision requirement diagram, which represents human decision making or automated decision making within a process. The main motivation of this work is the need to provide support to decision requirements, by offering adequate tools to the systems engineers that improve the design and handling of these types of requirements. Considering this, the approach presented in this paper is a step forward to support decision requirements, completing the different system engineering views by combining of SysML, BPMN and DMN.

III. BASIC CONCEPTS

This section presents the basic concepts on which the proposed approach is based. Section A describes the role of requirements in system engineering, Section B introduces traceability related concepts in SysML, Section C describes the SysML requirements diagram and its relationships with others diagrams like the: use case diagram, block definition diagram and state machine used in the approach. Section D shows some concepts about DMN and its relationship with BPMN.

A. Requirements

Requirements are the base in system development. They determine what the system has to offer, they can specify a desired feature, property, or behavior of a system, i.e., requirements set out what the system should do and define constraints that it has [20]. The concept of the requirement may also be further classified as [21]:

- Business Requirement. A Business Requirement is used to indicate the needs of a business. This impact on the organization and all the projects within it.
- Functional Requirement. Functional Requirements produce an observable result to someone, or something, that is using the system, i.e., they are the services that the system should provide.
- Non-functional Requirement. A Non-functional Requirement will constrain, or limit in some way, the way in which Functional Requirement may be realized.

B. Traceability in SysML

In [2], INCOSE indicates that “requirements traceability refers to the ability to describe and follow the life of a requirement in both a forward and backward direction along the design stages”. Traceability plays an important role as part of any MBSE methodology [3]. MBSE emphasizes the use of models to perform the systems engineering activities, as mentioned before. In fact, “MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” [22].

Modeling with SysML allows good traceability because it defines relationships between requirements and among other modeling elements [23][24]. Figure 1 describes the approach in which SysML accomplishes traceability by means of the 4 pillars presented in Section I. This figure shows the system model as an interconnected set of model elements. The arrows that cross the pillars, as seen in Figure 1, illustrate how the different elements belonging to the different types of diagrams that participate in the pillars are related, supporting requirements traceability.

C. SysML Requirements diagram and its relationships with others diagrams

In SysML, the requirements diagram shows the set of requirements and the relationship between them. A requirement specifies a function that must be satisfied or a condition that a system must achieve. Requirements modeling provides a bridge among different SysML diagrams because a requirement can appear on other diagrams to show its relationship to other modeling elements. The relationships that allow relating requirements with other requirements or with other modeling elements are [5]:

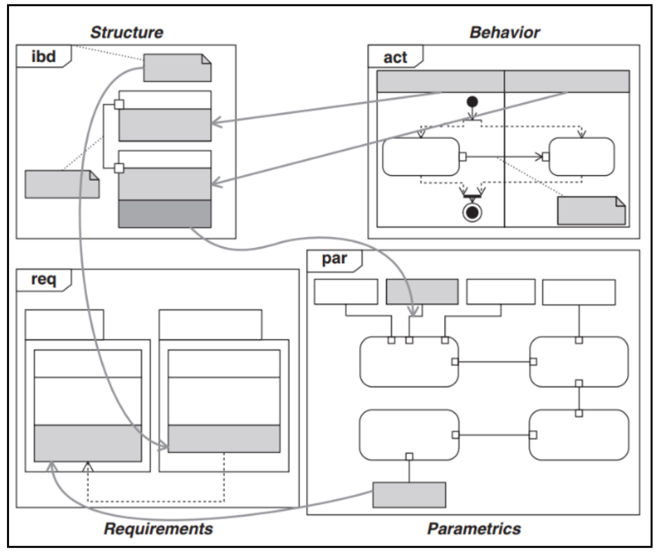


Figure 1. A system model example in SysML where requirements traceability is indicated with the connecting arrows (from [6]).

- Containment: a relationship, which is used to represent how a compound requirement can be partitioned into a set of simpler requirements (denoted graphically with a circle containing a + symbol).
- «deriveReqt»: a relationship, which describes that a requirement is derived from other requirement.
- «satisfy»: a relationship that describes that a design element satisfies a requirement. Usually, a requirement is satisfied by a block.
- «verify»: a relationship that connects a test case with the requirement that is verified by that test case.
- «refine»: a relationship, which specifies that a model element describes the properties of a requirement in more detail.
- «trace»: a general-purpose relationship between a requirement and any other model element.

The requirements can be related to the use cases through the relationship «refine». On the one hand, a use case can be viewed as functionality and/or capacity. On the other hand, a requirement specifies a capability or condition that must be satisfied, as previously mentioned, therefore, a use case diagram may be used to refine one or more functional requirements. In addition, the requirements are related to the blocks through the relationship «satisfy», as mentioned before. The block definition diagram captures the relation between blocks, such as a block hierarchy. Since the activities can be seen as a block, they can have associations between each other, including composition associations. Activities in block definition diagrams appear as regular blocks, except for the «activity» keyword [5]. Depending on the nature of the block, this can have a behavior associated, in that case, states machine can be used to describe its internal states. The state machine diagram is used to specify a behavior, with a focus on the set of states of a block and the possible transitions between those states in response to event occurrences, i.e., the state machine diagram presents behavior of an entity, as a block, in terms of its transitions between states triggered by events [25].

SysML enables characterization of any type of requirements for the system, including user, technical or others. A modeler can then define relationships between the specified requirements, providing the opportunity to create traceability among them. There is also an opportunity to create traceability from the logical and structural architecture design to their requirements, one of the most critical activities in systems engineering [26].

D. BPMN and DMN

The OMG provides the DMN notation for modeling decisions, which is not only understandable to stakeholders but it is also designed to be used in conjunction with the BPMN standard notation [12].

DMN provides constructs to both decision requirements and decision logic modeling. For decision requirements modeling, it defines the concept of Decision Requirements Graph (DRG) depicted with the Decision Requirements Diagram (DRD). This latter shows how a set of decisions depends on each other, on input data, and on business knowledge models. A decision element determines an output from the inputs, using decision logic, which may reference one or more business knowledge models. This denotes a function encapsulating business knowledge, e.g., as business rules, a decision table, or an analytic model. A decision table is a representation of decision logic, based on rules that determine the output depending on the inputs [12]. Decision-making modeled in DMN may be mapped to BPMN tasks or activities (Business Rules) within a process modeled with BPMN. The combined use of both thus provides a graphical language for describing decision-making, i.e., the BPMN tasks involving a decision can invoke a DMN decision model.

IV. MBSE and Requirements Traceability with SysML

In this section, our contribution of traceability of requirements using SysML, BPMN, and DMN is detailed. Section A presents an extension to SysML for BPMN tasks while Section B describes details on the proposed approach for requirements traceability.

A. SysML extensions for BPMN tasks

In order to support the modeling of BPMN tasks in SysML, the element of SysML block diagram must be extended. As noted above, in block definition diagrams, the activities appear as regular blocks with an «activity» stereotype. The stereotypes are one of the extensibility mechanisms of UML, therefore also of SysML, that enable to extend its vocabulary allowing the creation of new kinds that are derived from existing ones but specific to a problem [27]. Stereotypes are shown as text strings surrounded by the symbols "« »" [28]. The stereotypes change or add semantics to a base SysML element.

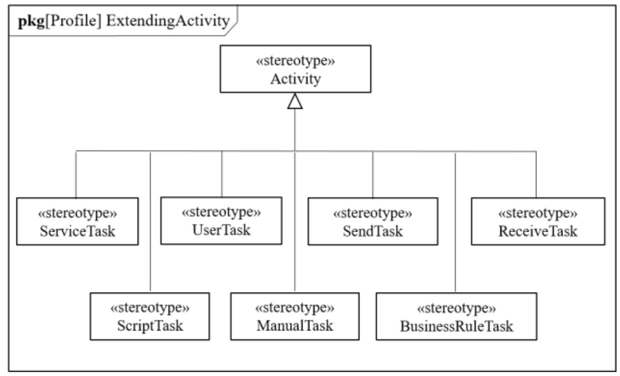


Figure 2. Extension of the SysML «Activity» stereotype.

Figure 2 shows the new types of activities through a generalization in order to support all types of BPMN tasks [5].

This extension consists of the following stereotypes:

- «serviceTask»: represents a task that uses a web service or an automated application.
- «sendTask»: represents a simple task that is designed to send a message to an external participant.
- «receiveTask»: represents a simple task that is designed to wait for a message to arrive from an external participant.
- «userTask»: represents a task where a person performs the task with the assistance of a software application.
- «manualTask»: represents a task that is expected to be performed without the aid of any business process execution engine or any application.
- «scriptTask»: represents a task executed by a business process engine.
- «businessRuleTask»: represents a task that involves decision-making.

The business rule task was defined in BPMN as a placeholder for (business-rule-driven) decisions, being the natural placeholder for a decision task [12].

B. Requirements Traceability using SysML and BPMN-DMN

The interaction between the process and the decision models plays a crucial role because a decision can affect the process behavior or flow [9]. Therefore, it is important that decision-making must be considered as a requirement that should be performed and satisfied.

The approach will be illustrated with an example intended to carry out the traceability of the requirements through forward engineering, mainly focusing on those requirements involved in the decision-making activities, in the blocks that have a behavior related with some of these activities, and in the use cases that refine some of those requirements, with the aim of integrating and covering their different views.

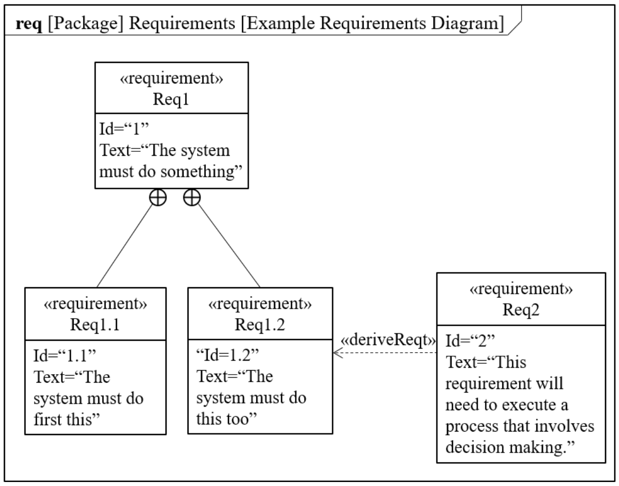


Figure 3. An example of a requirements diagram.

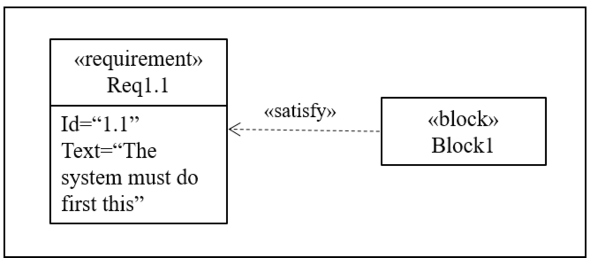


Figure 4. Example of a «satisfy» relationship between an Block and a Requirement.

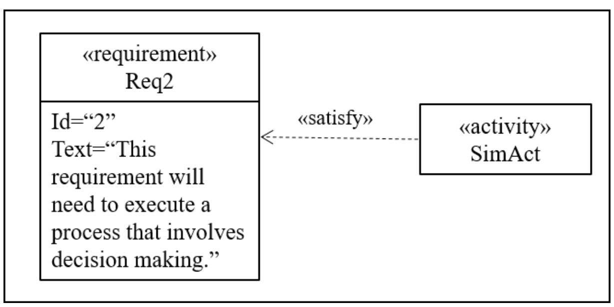


Figure 5. Example of a «satisfy» relationship between an Activity and a Requirement.

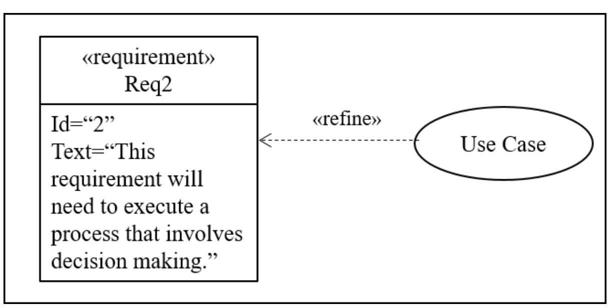


Figure 6. Example of a «refine» relationship between a Use Case and a Requirement

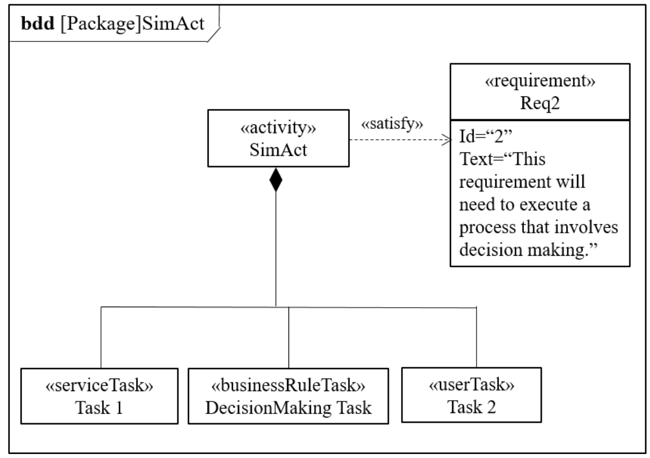


Figure 7. Block definition diagram with activities as blocks.

As it was mentioned before, the SysML requirement diagram has several relationships used to connect requirements. For example, Figure 3 presents a SysML requirements diagram labeled "*Example Requirements Diagram*", which shows the relationship between requirements. In particular, it can be observed that the requirement with *id="2"* has a relationship with the requirement with *id="1.2"* through the «deriveReqt» relation. This relation specifies that the requirement with *id="2"* is derived from the requirement with *id="1.2"*.

Requirements can be related to other requirements and to other modeling elements through a specific set of relationships as mentioned before. Relationships between requirements and other modeling elements can appear on various types of diagrams. Figure 4 shows an example of a «satisfy» relationship between a *Block1* block and a *Req1.1* requirement, Figure 5 presents an example of a «satisfy» relationship between a *SimAct* activity and the *Req2* requirement, while Figure 6 shows an example of a «refine» relationship between a *Use Case* and a *Req2* requirement. Both requirements appear in the requirement diagram presented in Figure 3.

The interpretation of the «satisfy» relationship is that the design of the activity depends on the requirement, meaning that if the requirement changes, the design of the activity must be changed. The interpretation of the «refine» relationship is that the Use Case is more concrete than the requirement, i.e., less abstract.

Once the main requirements have been captured, the elements responsible to satisfy those requirements are modeled through a block definition diagram. As previously mentioned, activities can be seen as a block, except for the «activity» keyword [5]. This later provides a means for representing activity decomposition and allows a requirement to be satisfied with an activity. This activity can then be implemented in terms of the decision requirements diagrams and decision tables DMN. Figure 7 shows an example of decomposition of the *SimAct* activity, which was presented in Figure 5 by using the stereotypes proposed in Section IV-A. The block definition diagram shown in this

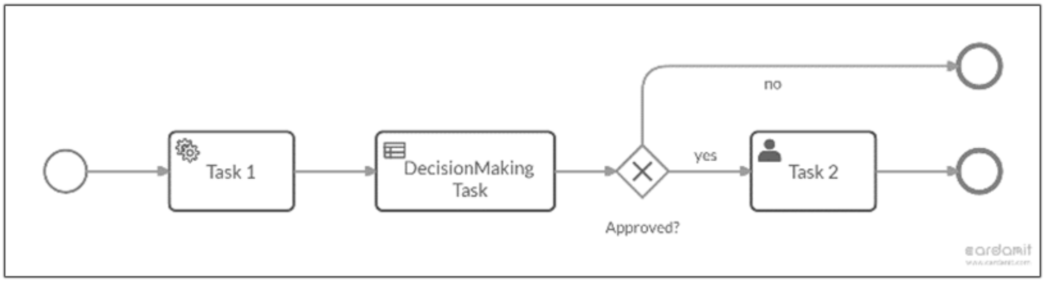


Figure 8. BPMN diagram of SimAct.

figure indicates that the *SimAct* is an activity composed of other activities, including *Task 1*, *DecisionMaking Task* and *Task 2*, all these activities being of BPMN activity types.

Finally, in order to cover the different views of the requirements, a BPMN model is constructed and associated to *SimAct* activity in order to show its behavior, as can be seen in Figure 8. In this figure, the activities that compose *SimAct*, which were modeled in Figure 7, are explicitated in BPMN format.

To conclude, the decision model related to the business rule task is built, since when BPMN and DMN are used, the BPMN tasks (business rule) have a link associated to the decision model, as mentioned in Section III-D. The *DecisionMaking Task* can then be implemented in terms of the associated decision requirements diagrams and decision tables.

V. CASE STUDY

To demonstrate our approach, we conducted a case study that includes the partial modeling of a Biodiesel Distiller and its requirements management. This case study illustrates how to carry out the traceability of the requirements through forward engineering.

Biodiesel is a type of biofuel that is similar to petroleum-based diesel, which can replace fossil fuel diesel. It is a sustainable fuel that is produced from fatty acids derived from animals such as beef fat, pork fat, chicken fat; and vegetable oils such as corn oil and cooking oil like those from restaurants that have already been used and disposed of. These oils are converted to diesel fuel through a chemical process.

Distilled biodiesel is a clean fuel that has been purified through the process of distillation. Biofuel distillation is a method that consists of taking a biofuel and removing particles and impurities within the liquid through an evaporation and condensation process.

The requirements diagram in Figure 9 shows the breakdown of the Biodiesel Distiller's requirements into a hierarchy of more refined requirements. This diagram named "Biodiesel Distiller Requirements Diagram" shows the relationship between its elements. In particular, it can be observed that the requirement *Initial Statement* is partitioned into a set of simpler requirements: *Generate Biodiesel*, *Heat Exchanger*, *Boiler*, *Biodiesel Properties* and *Distill Water*. In addition, two use cases can be seen in the figure: *Distill* and *Change temperature*, one of them refines *Generate Biodiesel* requirement, and the other refines *Heat Exchanger* requirement. In other words, both use cases are more concrete than the requirements, as was indicated in Section IV-B.

As previously mentioned, requirements can be related to other modeling elements through a specific set of relationships as «satisfy» relationship between a block and a requirement. Furthermore, activities can be seen as blocks. As mentioned in Section III-C, depending on its nature, a block can have a behavior associated and this behavior can be modeled using state machines.

Once the main requirements of *Biodiesel Distiller* have been captured, the elements responsible to satisfy them are modeled through a block definition diagram.

Figure 10 illustrates how Machine activity is composed of *Boiler* block, *Water* block, *Generator* activity, which satisfies *Generate Biodiesel* requirement, and *Heat Exchanger* block, which satisfies *Heat Exchanger* requirement, both requirements appear in the requirements diagram presented in Figure 9.

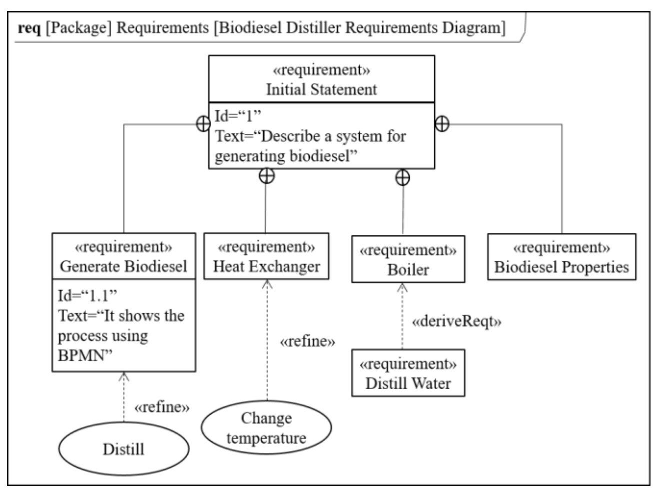


Figure 9. Requirements diagram: *Biodiesel Distiller Requirements Diagram*

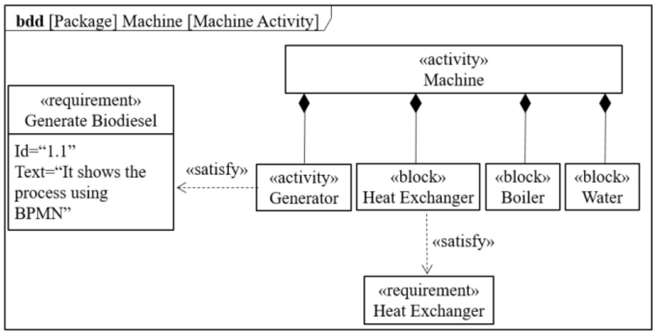


Figure 10. Generator activity satisfies the Generate Biodiesel requirement, and Heat *Exchanger* block satisfies the *Heat Exchanger* requirement

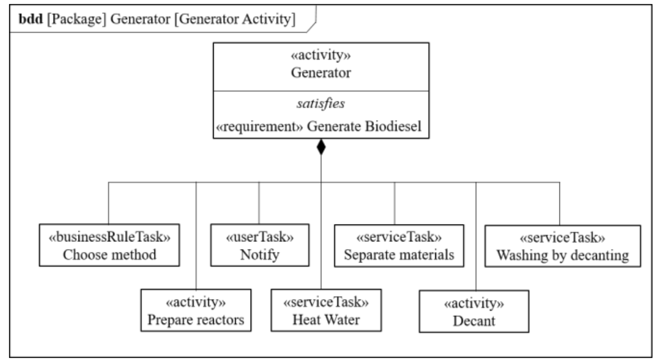


Figure 12. Decomposition of the *Generator* activity.

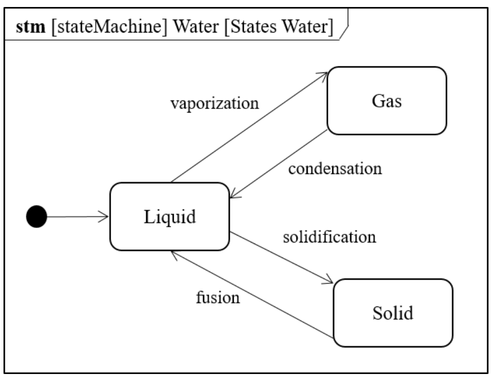


Figure 11. States machine diagram: *States Water*

U	material	Method
	Text [beef, pork, chicken, vegetable]	Text [M1,M2,M3,M4]
1	beef	M1
2	pork	M2
3	chicken	M3
4	vegetable	M4

Figure 13. Decision table to *Choose method*.

Continuing with the approach, the state machine diagram in Figure 11 shows possible states of Water block presented in Figure 10, which correspond to the states of water during the distillation process shown in Figure 14. Figure 12 illustrates the decomposition of the *Generator* activity by using the stereotypes proposed in Section IV-A. The block definition diagram shown in this figure indicates that the *Generator* is an activity composed of other activities such as: *Choose method* business rule, which involves decision-making, *Prepare reactors* activity, *Notify* user task, *Heat Water* service task, *Separate materials* service task, *Decant* activity and *Washing by decanting* service task, all these activities being of BPMN activity types.

Following the approach presented in this work, a BPMN model is constructed in order to cover the different views of the requirements. This model shows the process, which is carried out to generate biodiesel associated with the *Generator* activity. Its behavior can be observed in Figure 14. Note that the states of water presented in Figure 11 participate in several of the activities of the BPMN model such as: *Decant*, *Separate materials*, *Heat Water and Washing by decanting.*

The decision model related to the business rule task is built once the BPMN model is generated. To prepare the reactors, the type of method to be used must be known and this depends on the type of material that will be used for the generation of biodiesel. The materials can be beef fat, pork fat, chicken fat, and vegetable fat. In the case of study, this decision making is shown in terms of the decision table as shown in Figure 13.

As it was mentioned before, traceability refers to the capability to describe and follow the life of a requirement in both a forward and backward direction along the design stages. Traceability plays an important role in the MBSE methodology. To conclude the study case, and in order to show the requirements traceability presented in this article, Figure 15 shows our approach using three of the four pillars of SysML presented in Section I. This figure shows, through direct engineering, how traceability is carried out, showing how all the models and their elements presented in this section are connected to each other.

As it was mentioned before, the multiple cross-cutting relationships between the model's elements enable systems engineers to view several different perspectives that focus on different aspects of the system.

The arrows that cross the pillars structure, behavior, and requirements, as seen in Figure 15, illustrate how the different elements belonging to the different types of models that participate in these pillars are interconnected, supporting in this way requirements traceability.

Figure 15 shows the different diagrams presented in this study case: requirement diagram, block definition diagram, state machine diagram, BPMN diagram, and decision table, respectively.

The Requirements pillar contains the requirements model that was initially built for the development of the case study.

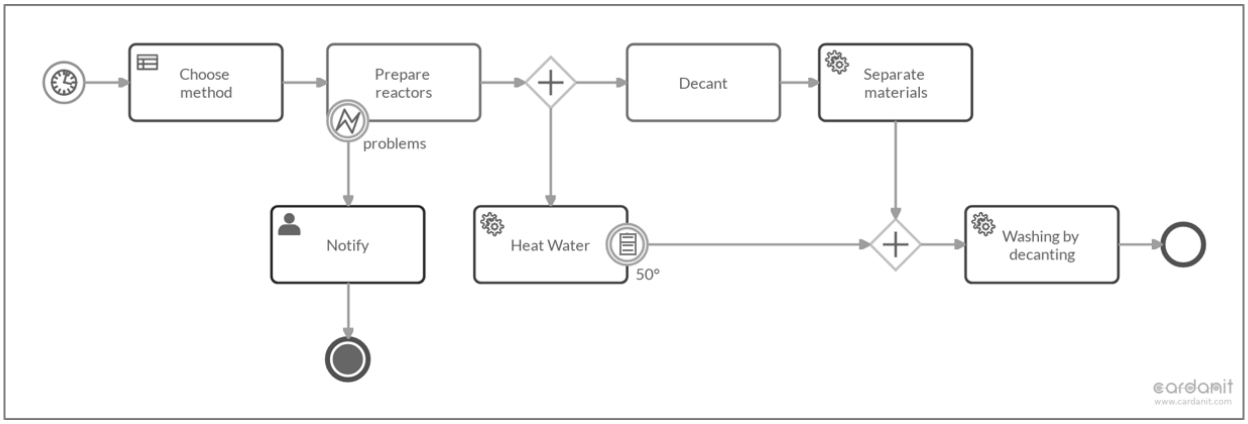


Figure 14. BPMN model of *Generator*.

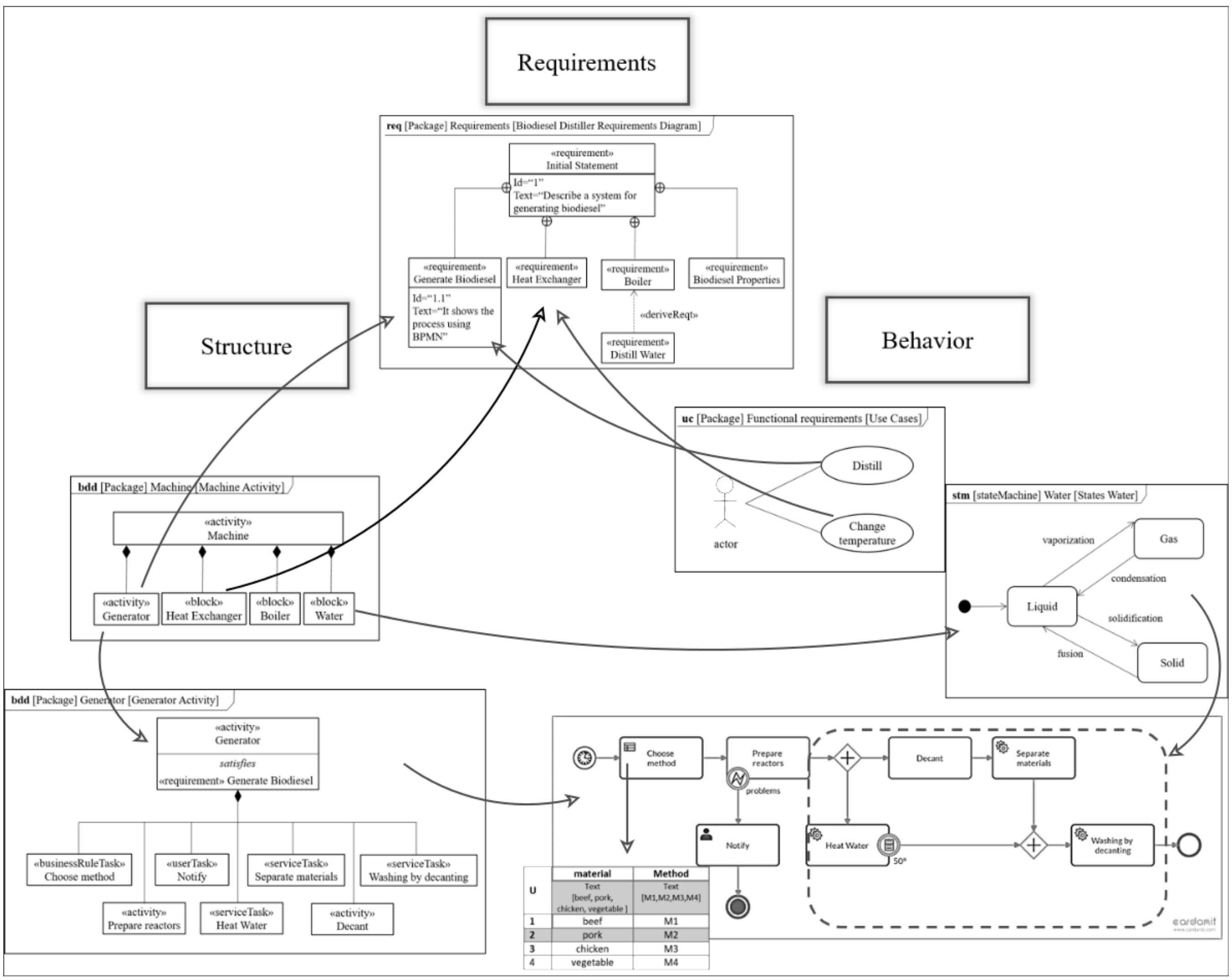


Figure 15. Traceability study case.

The Structure pillar contains two block definition diagrams: the *Machine Activity* block definition diagram and the *Generator Activity* block definition diagram. The *Machine Activity* block definition diagram has four relations. One of them shows the relationship between *Heat Exchanger* block and *Heat Exchanger* requirement through a *satisfy* relationship, as mentioned before. The other relation illustrates the relationship between *Generator Activity* and *Generate Biodiesel* requirement through a *satisfy* relationship, also this diagram has another relation toward *Generator Activity* block definition diagram, which shows how this activity is decomposed by using the extension proposed in this article. The fourth relation shows how *Water* block is related with states machine diagram in Behavior pillar. The *Generator Activity* block definition diagram has a relationship with BPMN model because of model illustrates the process, which is carried out to generate biodiesel associated with the *Generator* activity.

The Behavior pillar contains three diagrams. A use cases diagram shows the use cases that refine some requirements in the requirement diagram presented, in other words, *Distill* use case refines *Generate Biodiesel* requirement, and *Change temperature* use case refines *Heat Exchanger* requirement. The other diagram is *States Water* state machine diagram, which is used to describe the possible states of *Water* block in *Machine Activity* block definition diagram being these possible states used in some activities in BPMN model, as mentioned before. These activities are grouped in the model (rounded corner rectangle with a solid dashed line). Finally, this pillar has a BPMN model, which shows the process to generate biodiesel as can be observed in Figure 15 and mentioned before. In addition, the same figure shows the decision model related to the Choose method business rule task.

VI. CONCLUSION

Traceability plays an important role in any Model-Based Systems Engineering methodology. This methodology emphasizes the use of models to perform the systems engineering activities. The objective of this work has been to carry out requirements traceability understanding that requirements traceability is the capability to follow the life-cycle of the requirement. SysML is a general-purpose modeling language, based on UML, which enables traceability because it defines relationships between requirements and other modeling elements. The combination of SysML and BPMN-DMN is attractive and is a step forward that enhances the modeling of the different views of the system to be built including decision requirements. The proposed approach uses the definition of new stereotypes in SysML to support all types of BPMN tasks.

This proposal seeks to integrate and cover the different views of all requirements, helping systems engineers to improve the design of them.

The approach was illustrated through a case study to show how traceability of requirements can be performed.

In future work, we consider analyzing the link among the DMN decision requirements diagram, SysML requirements diagram, and use cases diagram. In addition, we consider working with parametric diagrams in order to complete the pillars of SysML.

ACKNOWLEDGMENT

The authors thank the reviewers of the ICSEA'19 conference and the IARIA Journal for the very useful comments that have contributed to enhance both the original and the extended versions of the paper.

REFERENCES

[1] C. Abdelahad, D. Riesco, and C. Kavka, "A SysML-based Approach to Requirements Traceability using BPMN and DMN," ICSEA, pp. 210-216, 2019.

[2] INCOSE https://www.incose.org/ [retrieved: December, 2020].

[3] J. Jacobs and A. C. Simpson, "Towards a process algebra framework for supporting behavioural consistency and requirements traceability in SysML," in Proceedings of the 15th International Conference on Formal Engineering Methods (ICFEM 2013), Lecture Notes in Computer Science, Springer, vol. 8144, pp. 265-280, 2013.

[4] OMG https://www.omg.org/ [retrieved: December, 2020].

[5] SysML https://www.omg.org/spec/SysML/1.6/PDF [retrieved: December, 2020].

[6] S. Friedenthal, A. Moore, and R. Steiner, "A Practical Guide to SysML: The Systems Modeling Language," Elsevier, 2014.

[7] P. Spoletini and A. Ferrari, "Requirements elicitation: a look at the future through the lenses of the past," In 2017 IEEE 25th International Requirements Engineering Conference (RE), IEEE, pp. 476-477, 2017.

[8] Modelica https://www.modelica.org/ [retrieved: December, 2020].

[9] Business process model and notation https://www.omg.org/spec/BPMN/2.0.2/PDF [retrieved: December, 2020].

[10] D. Q. Birkmeier, S. Klöckner, and S. Overhage, "An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users," ECIS 2010 Proceedings 51, 2010.

[11] C. V. Geambașu, "BPMN vs UML activity diagram for business process modeling," Accounting and Management Information Systems vol. 11, no. 4, pp. 934-945, 2012.

[12] Decision Model and Notation https://www.omg.org/spec/DMN/1.2/ [retrieved: December, 2020].

[13] E. J. Vidal and E. R. Villota, "SysML as a Tool for Requirements Traceability in Mechatronic Design," In Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering, ACM, pp. 146-152, 2018.

[14] Z. H. U. Shaofan, T. A. N. G. Jian, J. M. Gauthier, and R. Faudou, "A formal approach using SysML for capturing functional requirements in avionics domain," Chinese Journal of Aeronautics, vol. 32, no. 12, pp. 2717-2726, 2019.

[15] R. Baduel, M. Chami, J. M. Bruel, and I. Ober, "SysML Models Verification and Validation in an Industrial Context: Challenges and Experimentation," In European Conference on Modelling Foundations and Applications. Springer, Cham, pp. 132-146, 2018.

[16] A. Salado and P. Wach, "Constructing True Model-Based Requirements in SysML," Systems, vol. 7, no. 2, 21 pages, 2019.

[17] K. Gruber, J. Huemer, A. Zimmermann, and R. Maschotta, "Integrated description of functional and non-functional requirements for automotive systems design using SysML," 2017 7th IEEE Int. Conf. on System Engineering and Technology (ICSET), pp. 27-31, 2017.

[18] M. R. S. Marques, E. Siegert, and L. Brisolara, "Integrating UML, MARTE and SysML to improve requirements specification and traceability in the embedded domain," In 2014 12th IEEE International Conference on Industrial Informatics (INDIN), IEEE, pp. 176-181, 2014.

[19] H. Dubois, M. A. Peraldi-Frati, and F. Lakhal, "A model for requirements traceability in a heterogeneous model-based design process: Application to automotive embedded systems," In 2010 15th IEEE International Conference on Engineering of Complex Computer Systems, IEEE, pp. 233-242, 2010.

[20] I. Sommerville, "Software Engineering," Seventh Edition, Pearson Education, 2004.

[21] J. Holt and S. Perry, "SysML for systems engineering," vol. 7, IET, 2008.

[22] T. Weilkiens, "Systems engineering with SysML/UML: modeling, analysis, design," Elsevier, 2011.

[23] O. C. Z. Gotel and A. C. W. Finkelstein, "An Analysis of the Requirements Traceability Problem," Proc. IEEE Int. Conf. on Requirements Engineering, pp. 94-101, 1994.

[24] K. Hampson, "Technical evaluation of the systems modeling language (SysML)," Procedia Computer Science, vol. 44, pp. 403-412, 2015.

[25] L. Delligatti, "SysML distilled: A brief guide to the systems modeling language," Addison-Wesley, 2013.

[26] http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf [retrieved: December, 2020].

[27] UML 2.4 "Infrastructure Specification" https://www.omg.org/spec/UML/2.4.1/ [retrieved: December, 2020].

[28] G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language User Guide," Addison-Wesley, 1999.

Build Comparator: Integrated Semantic Comparison for Continuous Testing of Android-based Devices using Sandiff

Carlos E. S. Aguiar, Jose B. V. Filho, Agnaldo O. P. Junior,
Rodrigo J. B. Fernandes, Cícero A. L. Pahins, Paulo C. R. Fonseca
Sidia R&D Institute
Manaus, Brazil
Emails: {carlos.aguiar, jose.vilarouca, agnaldo.junior,
rodrigo.fernandes, cicero.pahins, paulo.fonseca}@sidia.com

***Abstract*—With ever-larger software development systems consuming more time to perform testing routines, it is necessary to think about approaches that accelerate continuous testing of those systems. This work aims to allow the correlation of semantic modifications with specific *test cases* of complex suites, and based on that correlation, skip time-consuming routines or mount lists of priority routines (*fail-fast*) to improve the productivity of mobile developers and time-sensitive project deliveries and validation. In order to facilitate continuous testing of large projects, we propose Sandiff, a solution to efficiently analyze semantic modifications on files that impact domain-specific testing routines of the official Android Test Suite. We also propose the *Build Comparator*, an integrated tool that leverages the *semantic comparison* on *real-world* use cases. We demonstrate our approach by evaluating both *semantic coverage* and scalability on a set of *commercially-available* Android images of a large mobile-related company that comprises both major and minor versions of the system.**

Keywords–Testing; Validation; Content Comparison; Continuous Delivery; Tool.

I. INTRODUCTION

As software projects grow up, continuous testing becomes critical, but at the same time, complicated and time-consuming. Consider a project with a million files and intermediate artifacts. A *test suite* that offers *continuous testing* functionalities must perform without creating bottlenecks or impacting project deliveries. However, effectively using continuous integration can be a problem: tests are time-consuming to execute. Consequently, it is impractical to run complete modules of testing on each build. In these scenarios, it is common that teams lack *time-sensitive* feedback about their code and compromise user experience.

The testing of large software projects is typically bounded to robust test suites. Moreover, the quality of testing and evaluation of ubiquitous software can directly impact people's life, a company's perceived image, and the relation with its clients. Companies inserted in the Global Software Development (GSD) environment, i.e., with a vast amount of developers cooperating across different regions of the world, tend to design a tedious testing and evaluation process that becomes highly time-consuming and impacts the productivity of developers. Moreover, continuous testing is a *de facto* standard in the software industry. During the planning of large projects, it is common to allocate some portion of the development period to design testing routines. Test-Driven Development (TDD) is a well-known process that promotes testing before feature development. Typically, systematic software testing approaches lead to computing and time-intensive tasks.

Sandiff [1] is a tool that helps to reduce the time spent on testing of large Android projects by enabling to skip domain-specific routines based on the comparison of meaningful data without affecting the functionality of the target software. For instance, when comparing two Android Open Source Project (AOSP) builds generated in different moments, but with the same source code, create the environment and build instructions, the final result is different in byte level (byte-to-byte). Still, it can be semantically equivalent based on its context (meaning). In this case, it is expected that these two builds perform the same. However, how to guarantee this? Our solution relies on how to compare and demonstrate that two AOSP builds are semantically equivalent. Another motivation is the relevance of Sandiff to the continuous testing area, where it can be used to reduce the time to execute the official Android Vendor Test Suite (VTS). As our solution provides a list of semantically equivalent files, it is possible to skip tests that show the behavior provided by these files. The Figure 1 shows the execution official Android Test Suite is execute in a *commercially-available* build based on AOSP. The execution of all modules exceeded 4 hours, compromising developer performance and deliveries on a planned schedule.

By *comparison of meaningful data*, we mean comparison of sensitive regions of critical files within large software: different from a byte-to-byte comparison, a semantic comparison can identify domain-related changes, i.e., it compares sensitive code paths, or *key-value* attributes that can be related to the output of large software. By *large*, we mean software designed by a vast number of developers inserted in a distributed software development environment; after that, automatic test suits are necessary.

Another motivation of Sandiff is to enable developers to find bugs faster on complex software. Take as an example a camera bug in which the component or module is part of a complex Android subsystem stack covering various architectural levels: application framework, vendor framework, native code, firmware, and others. With the help of the Sandiff, a developer can analyze the semantic comparison between different software releases and narrow the source of the problem.

In summary, we present the key research contributions of our proposal:

Suite/Plan	VTS/VTS
Suite/Build	9.0_R9 / 5512091
Host Info	seltest-66 (Linux - 4.15.0-51-generic)
Start Time	Tue Jun 25 16:17:23 AMT 2019
End Time	Tue Jun 25 20:39:46 AMT 2019
Tests Passed	9486
Tests Failed	633
Modules Done	214
Modules Total	214
Security Patch	2019-06-01
Release (SDK)	9 (28)
ABIs	arm64-v8a,armeabi-v7a,armeabi

Figure 1. Summary of the official Android Test Suite – *Vendor Test Suite* (VTS) – of a *commercially-available* AOSP build.

1. An approach to perform *semantic* comparison and facilitate *continuous testing* of large software projects.
2. An integrated *Build Comparator* tool that leverages semantic comparison to support Android-based software releases and DevOps teams.
3. An evaluation of the impact of using Sandiff in real-world and *commercially-available* AOSP builds.

Our paper is organized as follows. In Section II, we provide an overview of *binary* comparators and their impact on continuous testing of large projects. In Section III, we describe Sandiff and its main functionalities: (i) input detection, (ii) content recognition, and (iii) specialized semantic comparison. In Section IV, we present the *Build Comparator* tool and its architecture, which enables the integration of Sandiff on systems of the Android development environment. In Section V, we present the evaluation of Sandiff in *commercially-available* builds based on AOSP and discuss the impact of continuous testing of those builds. We conclude the paper with avenues for future work in Section VI.

II. Related Work

To the best of our knowledge, few literature approaches propose *comparison* of files with different formats and types. Most comparison tools focus on the comparison based on diff (text or, at most, byte position). Araxis [2] is a well-known commercial tool that performs three types of file comparison: text files, image files, and binary files. For image files, the comparison shows the pixels that have been modified. For binary files, the comparison is performed by identifying the differences in a byte level. Diff-based tools, such as Gnu Diff Tools [3] *diff* and *cmp*, also perform file comparison based on byte-to-byte analysis. The main difference between *diff* and *cmp* is the output: while *diff* reports whether files are different, *cmp* shows the offsets, line numbers and all characters where compared files differs. *VBinDiff* [4] is another diff-inspired tool that displays the files in hexadecimal and ASCII, highlighting the difference between them.

Other approaches to the problem of file comparison, in a *semantic* context, typically use the notion of *change or edit distance* [5] [6]. Wang et al. [5] proposed X-Diff, an algorithm that analyses the structure of an XML file by applying standard *tree-to-tree* correction techniques that focus on performance. Pawlik et al. [6] also propose a performance-focused algorithm based on the edit distance between ordered labeled nodes of an XML tree. Both approaches can be used by Sandiff to improve its XML-based semantic comparator. Similarly, with study applied on music notations, in [7] implemented a solution to compare files, like XML, based on a tree representation of music notation combining with sequence and tree distance. Besides, the authors show a tool to visualize the differences side-by-side.

In [8], is showed a tool named Diffi, which is a diff improved build to observe the correlation between file formats. So, Diffi verifies the heaps of the files' reflection levels and discovers which file levels can recognize the delta between two files correctly.

In [9], the authors explore the use of wavelets for the division of documents into fragments of various entropy levels, which made a separation between grouping sections to decide the similarity of the files, and finally, detect malicious software. Additionally, with a similar objective in [10] is investigated the applicability of machine learning techniques for recognizing criminal evidence in activities into file systems, verifying possible manipulations caused by different applications.

Different from previous works, the Sandiff also supports byte-level and semantic comparison simultaneously. However, the semantic comparison is the main focus of the tool to facilitate extensive software projects testing since it allows to discard irrelevant differences in the comparison.

III. Sandiff

Sandiff aims to compare meaningful data of two artifacts (e.g., directories or files) and report a compatible semantic list that indicates modifications that can impact the output of domain-related on continuous testing setups of large projects. In the context of software testing, syntactically different (byte-to-byte) files can be semantically equivalent. Once the characteristics of a context are defined, previously related patterns to this context can define the compatibility between artifacts from different builds. By definition, two artifacts are compatible when the artifact A can replace the artifact B without losing its functionality or changing their behavior. As each *file type* has its own set of attributes and characteristics, Sandiff employs specialized semantic comparators that are designed to support nontrivial circumstances of domain-specific tests. Consider the comparison of AOSP build output directory and its files. Note that the building process of AOSP in different periods of time can generate *similar* outputs (but not byte-to-byte equivalent). Different byte-to-byte artifacts are called syntactically dissimilar and typically require validation and testing routines. However, in the context where these files are used, the position of *key-value* pairs do not impact testing either software functionality. We define these files as *semantically compatible*, once Sandiff is able to identify them and suggest a list of tests to skip. Take Figure 3 as example. It shows a difference in the position of the last two lines. When comparing them byte-to-byte, this results in syntactically different files. However, in the execution context where these files are used, this is irrelevant, and the alternate position of lines does not change how the functionality works. Thus, the files are semantically compatible.

Sandiff consists of three main functionalities: (i) input detection, (ii) content recognition, and (iii) specialized semantic comparison, as shown in Figure 2. During analysis of

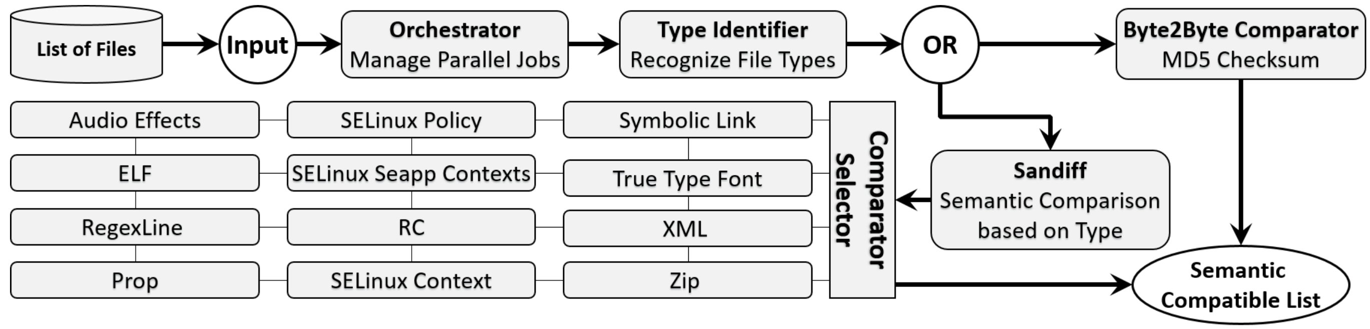


Figure 2. Sandiff verifies the semantic compatibility of two files or directories (list of files) and report their differences.

Configuration 1	*Configuration 2*
ro.build.version.preview_sdk=0	ro.build.version.preview_sdk=0
ro.build.version.codename=REL	ro.build.version.codename=REL
ro.build.version.all_codenames=REL	ro.build.version.all_codenames=REL
ro.build.version.release=8.0.0	ro.build.version.security_patch=17-08-05
ro.build.version.security_patch=17-08-05	ro.build.version.release=8.0.0

Figure 3. Example of AOSP configuration files.

TABLE I. SUMMARY OF *CONTENT RECOGNITION* ANALYSIS FOR EACH FILE.

Attribute	Meaning
Tag	Represents a file type
Action	Action to be taken with the file. (COMPARE *or* IGNORE)
Reason	In case of action IGNORE, the reason of ignore
Context	Information about context that is used to define the ACTION

directories and files, we can scan *image files* or *archives* that require particular *read* operations. The first step of Sandiff is to identify these files to abstract *file systems* operations used to access the data. This task is performed by the *Input Recognizer*. Then, the *Content Recognizers* and *Comparators* are instantiated. In order to use the correct *Comparator*, Sandiff implements *recognizers* that are responsible to detect supported *file types* and indicate if a file should be ignored or not based on a test context. Once Sandiff detects a valid file, it proceeds to the semantic comparison. The *Comparators* are specialized methods that take into consideration features and characteristics that are able to change the semantic meaning of execution or testing, ignoring irrelevant syntactical differences. Note that the correct analysis of semantic differences is both *file type* and context-sensitive. Sandiff implements two operation modes: (i) file and (ii) directory-oriented (walkables). In *file-oriented* mode, the input is two valid comparable files, whereas *directory-oriented* is the recursive execution of *file-oriented* mode in parallel, using a mechanism called *Orchestrator*. In the following sections, we describe the functionalities of Sandiff in detail.

A. Content Recognition

The Sandiff performs the analysis of file contents by leveraging internal structures, and known patterns to allow the correct selection of *semantic comparators*, i.e., artifact extension, headers, type signatures, and internal rules of AOSP to then summarize the results into (i) tag, (ii) action, (iii) reason, and (iv) context attributes, as shown in Table I. Each attribute helps the *semantic comparators* to achieve maximum semantic coverage over file types inside images embedded on commercially-available devices. The Android-based devices include several partitions that serve for specific purposes on boot process and general operation, as defined below:

- `system.img`: contains the Android operating system framework.
- `vendor.img`: contains any manufacturer-specific binary file that is not available to the AOSP.
- `userdata.img`: contains user-installed data and applications, including customization data.

To measure the semantic coverage, we gathered the percentage (amount of files) of file types inside `vendor.img`. We created a priority list to develop semantic comparators, as shown in the Table II. For instance, both ELF (32 and 64 bits) files represent about roughly 60% of total files inside `.img` files, whereas symbolic link files about 14% and XML files about 6%. This process enables us to achieve about 90% of semantic coverage. As the comparison is performed in a semantic form, it is necessary to know the context in which the artifact was used to enable the correlation between files and test cases. Note that a file can impact one or more tests in a different manner, e.g., *performance*, *security* and *fuzz* tests. The remaining 10% of files are compared using the byte-to-byte comparator.

Each *recognizer* returns a unique tag from a set of available tags or a tag with no content to indicate that the file could not be recognized. Recognizers can also decide whether a file should be ignored based on context by using the *action attribute* and indicating a justification in the *reason attribute*. Recognizers are evaluated sequentially. The first recognizer runs and tries to tag the file: if the file cannot be tagged, the next recognizer in the list is called, repeating this process until a valid recognizer is found or, in the latter case, the file is tagged to the *default comparator* (byte-to-byte). Table III summarizes the list of AOSP-based recognizers supported by Sandiff.

B. Semantic Comparators

Sandiff was designed to maximize semantic coverage of the AOSP by supporting the most relevant intermediate files used for packing artifacts into `.img` image files, i.e., the bootable binaries used to perform a factory reset and restore the original operating system of AOSP-based devices. To ensure

TABLE II. SUMMARY OF SANDIFF SEMANTIC COVERAGE.

File Type	# Files	Percentage (%)
ELF-64	320	31.34
ELF-32	298	29.19
Symbolic link	152	14.89
XML document text	63	6.17
RC	34	3.33
.bin	34	3.33
.tlbin	28	2.74
.prop	21	2.06
.conf	10	0.98
ASCII text	8	0.79
Exported SGML	6	0.59
.dat	6	0.59
.hcd	4	0.39
JAR	3	0.29
.txt	3	0.29
.xml	2	0.20
Gzip compressed data	1	0.10
SE Linux	1	0.10
PEM certificate	1	0.10

* **Green** = Semantic comparison is performed. **Red** = Semantic comparison not applicable. Default comparator (checksum) is performed. **Orange** = Semantic comparison not supported by Sandiff. Default comparator (checksum) is performed.

TABLE III. LIST OF AOSP-BASED *RECOGNIZERS* SUPPORTED BY SANDIFF.

Recognizer	Tags	Action
IgnoredByContextRecognizer	ignored_by_context	Ingore
ContextFileRecognizer	zip_manifest	Compare
MagicRecognizer	elf, zip, xml, ttf, sepolicy	Compare
AudioEffectsRecognizer	audio_effects_format	Compare
SeappContextsRecognizerc	seapp_contexts	Compare
PKCS7Recognizer	pkcs7	Compare
PropRecognizer	prop	Compare
RegexLineRecognizer	regex_line	Compare
SEContextRecognizer	secontext	Compare
ExtensionRecognizer	Based on file name. e.g.: file.jpg ? ”jpg”	Compare

the approach assertiveness, we performed an exploratory data analysis over each file type and use case to define patterns of the context's characteristics for each semantic comparator. The exploratory data analysis over each file type relies on three steps:

1. file type study;
2. where these files are used;
3. how these files are used (knowledge of its behavior).

The result of this analysis was used to implement each semantic comparator. The following subsections describe the main semantic comparators of Sandiff.

1) Default (Fallback) Comparator: Performs byte-to-byte (checksum) comparison and is the default comparator for binary files (e.g., *.bin*, *.tlbin*, and *.dat*). Acts as a *fallback* *alternative*, performing comparison for cases where (i) file type is not recognized or is unsupported and (ii) due to any errors during the comparison (e.g., corrupted or malformed files). Sandiff employs the industry standard [11] MD5 checksum algorithm, as it balances performance and simplicity to verify data integrity. For security-critical scenarios, Sandiff offers a set of *SHA* algorithms with improved robustness and reliability for collision attacks, i.e., verification of intentional corruption, despite lower run-time performance. The supported alternatives are: SHA1 [12], SHA224 [13], SHA256 [14], SHA384 [15], and SHA512 [16]. To choose the most suitable algorithm, it is necessary to consider the infrastructure, the application requirements and the knowledge of the developers. A complete overview and discussion about SHA algorithms is provided in [17], [18] reviews.

TABLE IV. EXAMPLE OF COMPARING TWO SEQUENCES OF BYTES.

Position	0	1	2	3	4	5	6	7	8
Sequence #1	1D	E1	2A	DD	5F	AE	F8	5F	19
Sequence #2	1D	ED	31	9E	5F	08	F8	5F	2E

When comparing two or more sequences of bytes, each position is represented on hexadecimal format, i.e., positional format that represents numbers using a base of 16 and uses symbols `0-9` and `A-F`, representing each byte by two hexadecimal digits. Table IV illustrates how Sandiff performs the byte-to-byte comparison. Note that in this case, the comparison verified that positions 1-3, 5 and 8 are different, summarizing the result as illustrated in Listing 1.

```
Difference(s):
    Differs on range between position 1 and 3
    Differs on byte at position 5
    Differs on byte at position 8
```

Listing 1. Example of checksum comparator output.

2) Audio Effects: AOSP represents audio effects and configurations in `.conf` files that are similar to `.xml`:

(i) `<name>{[sub-elements]}`
(ii) `<name> <value>`

Audio files are analysed by an ordered model detection algorithm that represents each element (and its sub-elements) as nodes in a tree alphabetically sorted.

3) Executable and Linking Format (ELF): ELF files are common containers for binary files in Unix-base systems that packs object code, shared libraries, and core dumps. This comparator uses the definition of the ELF format (`<elf.h>` library) to analyse (i) the files architecture (32 or 64-bit), (ii) the object file type, (iii) the number of section entries in header, (iv) the number of symbols on *.symtab* and *.dynsym* sections, and (v) the mapping of segments to sections by comparing program headers content.

To correlate sections to *test cases*, Sandiff detects semantic differences for AOSP *test-sensitive* sections (e.g., *.bss*, *.rodata*, *.symtab*, *.dynsym*, *.text*), listing and performing byte-to-byte comparison on all relevant ELF sections found on target files. Table VI summarizes irrelevant sections to ignore when implementing the semantic comparison for ELF files. When ELF files are Linux loadable kernel modules (`.ko` extension,

TABLE V. LIST OF LISTCOMPARATOR OPERATIONS MODES AND FLAGS SUPPORTED BY SANDIFF.

Flag	Description
File Order	Indicates if there are difference among the item position in list and the line number of file
Line Type	Indicates if file lines has only one key ONLY_KEY) or are (composed by more elements (MORE_THAN_KEY), with a value associated to the key
Displacement	Indicates if the result refers to differences semantically relevant (true), or differences semantically irrelevant (false)

TABLE VI. LIST OF IRRELEVANT SECTIONS FOR SEMANTIC COMPARISON OF ELF FILES.

Section	Reason
.debug_*	holds information for symbolic debugging
.comment	version control information
.gnu_debugdata	allows adding a subset of full debugging info to a special section of the resulting file so the stack traces can be more easily ”symbolicated”
.gnu_debuglink	contains a file name and a CRC checksum of a separated debug file
.gnu_hash	allow fast lookup up of symbols to speed up linking
.ident	where GCC puts its stamp on the object file, identifying the GCC version which compiled the code
.shstrtab	section names
.got.*	provides direct access to the absolute address of a symbol without compromising position-independence and sharebility
.note.gnu.build-id	unique identification 160-bit SHA-1 string

kernel object), the comparator checks if the module signature is present to compare its size and values. Signature differences are not considered relevant to semantic comparison. In case of any ELF file is corrupted or malformed, the *fallback* comparison is performed.

4) ListComparator: Base comparator for files structured as item lists, reporting (i) items that exist only in one of the compared files, (ii) line displacements (i.e., lines in different positions), (iii) comments and empty lines, and (iv) duplicated lines. To facilitate the correlation between files and *test cases*, Sandiff implements specific *list-based* semantic comparators for *Prop*, *Regex Line* and *SELinux* files, as they contain properties and settings that are specific to a particular AOSP-based device or vendor. To support such variety of files, the *list-based* comparators offers a list of operation modes to tackle specific scenarios, e.g., when the file has empty lines or comments semantically irrelevant, as summarized in Table V. The following paragraphs describe the *list-based* comparators of Sandiff.

a) Prop: Supports files with `.prop` extensions and formatted as `<key> = <value>` patterns. Prior to analysis, each line of a *.prop* file is categorized in *import*, *include* or *property*, as defined below:

1. *import*: lines with format `import <key>`, i.e., lines containing the word ”import”, followed by one key.
2. *include*: lines with format `include <key>`, i.e., lines containing the word ”include”, followed by one key.
3. *property*: lines with format `<key> = [<value>]`, i.e., lines containing a pair composed by a key and an associated value (optional) - separated by ”=” symbol.

After categorization, each line is parsed and added to its respective list, i.e., *import*, *include*, and *property* lists. Each of the three lists is individually compared with the others, generating disjoint results that are later jointed for reporting.

TABLE VII. LIST OF SEMANTIC IRRELEVANT PROPERTIES OF ANDROID BUILDS.

Property	Description
BD	Found in system/sepolicy_version
ro.bootimage.build.*	Build property set by android/build/make/core/Makefile
ro.build.*	Build property set by android/build/make/tools/buildinfo.sh at each new build
ro.expect.recovery_id	Build property set by android/build/make/core/Makefile
ro.factory.build_id	Build property set by android/build/make/core/main.mk
ro.ss.bid	Build property set by android/build/make/core/main.mk
ro.vendor.build.*	Build property set by android/build/make/core/Makefile

The *PropComparator* also provides a list of properties to be discarded (considered irrelevant) on the semantic comparison, as summarized in Table VII. A line can be ignored if is empty or commented.

b) RegexLine: Performs the comparison of files in which all lines match a user-defined regex pattern, e.g., '`/system/.`' or '`.so`', offering the flexibility to perform semantic comparison of unusual files.

c) SELinux: Security-Enhanced Linux, or SELinux, is a mechanism that implements Mandatory Access Control (MAC) in Linux kernel to control the permissions a subject context has over a target object, representing an important security feature for modern Linux-based systems. Sandiff supports semantic comparison of SELinux specification files that are relevant to *security test cases* of the VTS suite, i.e., *Seapp contexts*, *SELinux context*, and *SELinux Policy*, summarizing (i) components, (ii) type enforcement rules, (iii) RBAC rules, (iv) MLS rules, (v) constraints, and (vi) labeling statements.

5) RC: The Android Init System is responsible for the AOSP bootup sequence, *init* and *init* resources, components that are typically customized for specific AOSP-based devices and vendors. The initialization of modern systems consists of several phases that can impact a myriad number of *test cases* (e.g., *kernel*, *performance*, *fuzz*, *security*). Sandiff supports the semantic comparison of `.rc` files that contain instructions used by the *init* system:

- *imports*: analyses the importing calls of the invoking mechanism.
- *actions*: compares the sequence of `commands` to test if logical conditions are met, since commands in different order may lead to different results.
- *services*: tackles operation modes by analysing options (e.g., *oneshot*, *onrestart*, etc.) and characteristics (e.g., critical, priority, etc.) of the *init-related* services.

6) Symbolic Link: The semantic comparison of symbolic links is an important feature of Sandiff that allows correlation between *test cases* and absolute or relative paths that can be differently stored across specific AOSP-based devices or vendors, but result in the same output or execution. The algorithm is defined as follows: first it checks if the file status is a symbolic link, and if so, reads where it points to. With this content it verifies if two compared symbolic links points to same path. The library used to check the file status depends on the input type and is abstracted by *Input Recognizers*. The libraries are:

File System → `<sys/stat.h>`
Image File → `<ext2/ext2fs.h>`
ZIP → `<zip.h>`

7) True Type Font: Sandiff uses the Freetype library [19] to extract data from TrueType fonts, which are modeled in terms of faces and tables properties. For each property field, the comparator tags the *semantically* irrelevant sections to ignore during semantic comparison. This is a crucial feature of Sandiff since is common that vendors design different customization on top of the default AOSP user interface and experience.

8) XML: XML is the *de facto* standard format for web publishing and data transportation, being used across all modules of AOSP. To support the semantic comparison of XML files, Sandiff uses the well-known *Xerces* library [20] by parsing the Document Object Model (DOM), ensuring robustness to complex hierarchies. The algorithm compares nodes and checks if they have (i) different attributes length, (ii) different values, (iii) attributes are only in one of the inputs, and (iv) different child nodes (added or removed).

9) Zip and Zip Manifest: During the building process of AOSP images, zip-based files may contain Java Archives (`.jar`), Android Packages (`.apk`) or ZIP files itself (`.zip`). As these files follow the ZIP structure, they are analysed by the same semantic comparator. Note that, due to the *archive* nature of ZIP format, Sandiff covers different cases:

1. *In-place*: there is no need to extract files.
2. *Ignore Metadata*: ignore metadata that is related to the ZIP specification, e.g., archive creation time and archive modification time.
3. *Recursive*: files inside ZIP are individually processed by Sandiff, so they can be handled by the proper *semantic comparator*. The results are summarized to represent the analysis of the zip archive.

Another important class of files of the AOSP building process are the *ZIP manifests*. Manifest files can contain properties that are time-dependent, impacting *naive* byte-to-byte comparators. Sandiff supports the semantic comparison of *manifests* by ignoring *header* keys entries (e.g., String: ”`Created-By`”, Regex: ”`(.+)-Digest`”) and *files* keys entries (e.g., `SEC-INF/buildinfo.xml`).

Each *APK* – the package file format used by Android – has a set of manifest information and other metadata that are part of the signature process. The most relevant for semantic comparison are `META-INF/CERT.SF` and `MANIFEST.MF` files, since it contains integrity checks for files which are included in the distribution, as shown in Listings 2 and 3. As both files share the same structure, it is possible to analyse them with the same semantic comparator. The *ZipManifestComparator* parses both files and compares headers and digests by ignoring irrelevant header and files entries.

```
Manifest-Version: 1.0
Created-By: singlejar
Name: AndroidManifest.xml
SHA-256-Digest: cEnjm4r95tb8NSMCP6B2Nn+P1G8sIpeXpPtsmuvSnfM=
Name: META-INF/services/
     com.google.protobuf.GeneratedExtensionRegistryLoader
SHA-256-Digest: AT7RUk9qflHB8mVVceY0Zi7UuRK2bIPMewdxqL2zIBY=
Name: android-support-multidex.version.txt
SHA-256-Digest: OuJR1NnX1srJFP8Td2Bv9F5nMX3O5iAgxf15egCfa+Q=
```

Listing 2. The `MANIFEST.MF` file contains metadata used by the java run-time when executing the *APK*.

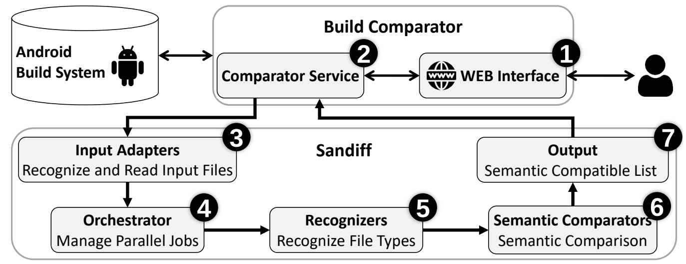


Figure 4. *Build Comparator* architecture and its relation with Sandiff's semantic comparison features.

```
Signature-Version: 1.0
Created-By: 1.0 (Android SignApk)
SHA-256-Digest-Manifest: rk0ZXezaawnGF65RmyYEmpqL+
    gFdHzRTNb3kr/NeNNQ=
X-Android-APK-Signed: 2, 3
Name: AndroidManifest.xml
SHA-256-Digest: ZgWkXiulhWzT7qwbAVYgepd5tyGt6D+RQNAeT+AJw1Y=
Name: META-INF/services/
    com.google.protobuf.GeneratedExtensionRegistryLoader
SHA-256-Digest: ASo5NB1Aa4gclvZke+olzjfErZMzxn/hthDK7Ann56w=
Name: android-support-multidex.version.txt
SHA-256-Digest: 6/lnFOH7mFVER94rAWcUmubglFFrHR7nf8+7zqQOgQs=
```

Listing 3. The `CERT.SF` file contains the whole-file digest of the `MANIFEST.MF` and its sections.

10) PKCS7: Public Key Cryptography Standards, or PKCS, are a group of public-key cryptography standards that is used by AOSP to sign and encrypt messages under a Public Key Infrastructure (PKI) structured as *ASN.1* protocols. To maximize semantic coverage, Sandiff ignores *signatures* and compares only valid *ASN.1* elements.

C. Orchestrator

The *Orchestrator* mechanism is responsible to share the resources of Sandiff among a variable number of competing comparison jobs to accelerate the analysis of large software projects. Consider the building process of AOSP. We noticed that, for regular builds, around 384K intermediate files are generated during compilation. In this scenario, running all routines of the official Android Test Suite, known as *Vendor Test Suite* (VTS), can represent a time consuming process that impacts productivity of mobile developers. To mitigate that, the *Orchestrator* uses the well-known concept of *workers* and *jobs* that are managed by a priority queue. A *worker* is a thread that executes both recognition and comparison tasks over a pair of files, consuming the top-ranked files in the queue. To accelerate the analysis of large projects, Sandiff adopts the notion of a *fail greedy* sorting metric, i.e., routines with higher probability of failing are prioritized. The definition of *failing priority* is context-sensitive, but usually tends to emphasize critical and time-consuming routines. After the processing of all files, the results are aggregated into a structured report with the following semantic sections: (i) addition, (ii) removal, (iii) syntactically equality, and (iv) semantic equality.

IV. Integrated Build Comparator Tool

To provide *semantic comparison* benefits on supporting software releases and DevOps operations, we propose *Build Comparator*, an integrated tool that abstracts configuration, image management, test execution, and report visualization to

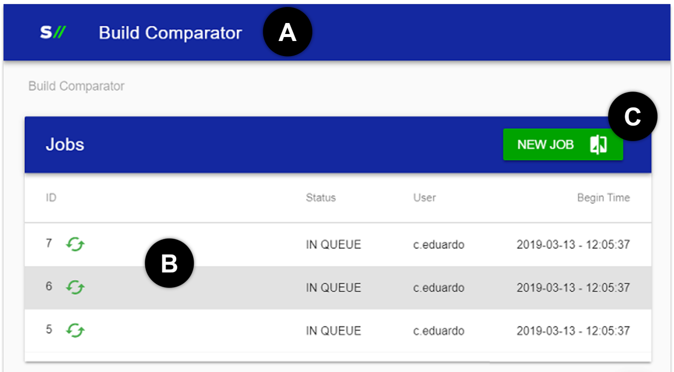


Figure 5. **(A)** Main *Build Comparator* interface. **(B)** List of jobs with unique identifier (ID), status, user, and job begin time. **(C)** Creation of comparison jobs based on (i) build systems, (ii) remote or (iii) local images.

facilitate Android-based development pipelines and processes. The *Build Comparator* can be integrated with systems that are commonly used for compilation jobs and managing the configuration and execution of Android's platform scripts, as shown in Figure 4. The proposed integration covers all main steps from user input to report visualization, as follows:

1. The user interacts with the tool through the **Web Interface**, which is responsible for providing communication between the user and service.
2. The **Comparator Service** provides a REST-based service that is responsible for managing the whole life-cycle of scheduled jobs (i.e., creation, management and running) using Sandiff as back-end. The *service* is also integrated with the systems responsible for compilation jobs, performing the user credential and image downloads.
3. On Sandiff side, the **Input Adapters** provide the ability to R/W different input files, abstracting the methods used to access the data.
4. The **Orchestrator** is responsible for managing the parallel jobs when Sandiff is filled with directories analysis, tracking file additions, removals, and type changes.
5. The **Recognizers** are responsible for determining the correct **Semantic Comparator** by analysing the (i) file type, (ii) header, and (iii) general structure.
6. In the last step, Sandiff generates the *semantic* compatible list, making it available to the *Comparator Service* and *Web Interface*.

In Figures 5 and 6, we summarize the main *Build Comparator* interfaces. We use a client-server architecture for the current implementation. The browser-based client is written in *JavaScript*, *HTML5*, and *Angular*. The server is a C++11 template-based implementation that exposes its API for queries via *HTTP*, enabling the comparison of Android builds according with the source of the artifacts. The architecture supports the most common integration methods with *continuous testing* tools or development pipelines:

- **Build Systems:** the artifacts are located in continuous integration and deployment servers, e.g., *QuickBuild* [21].
- **Remote:** the artifacts are located in an *HTTP* server or the cloud.
- **Local:** the artifacts are located in the user's personal computer.

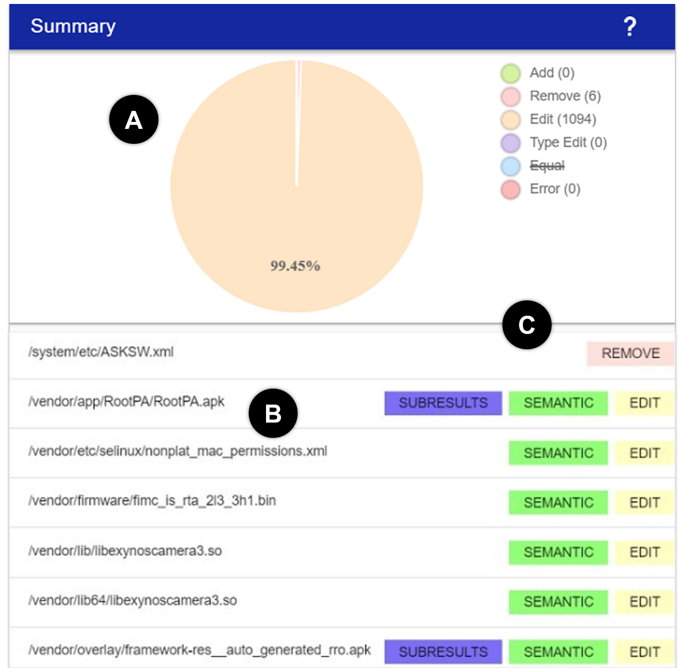


Figure 6. Interface that summarizes the *semantic comparison* results. **(A)** General statistics of the analysed pair of Android images. **(B)** List of semantic relevant artifacts that are supported by Sandiff. **(C)** Type of each semantic modification (add, remove, edit, type edit).

V. EXPERIMENTS

A. *Semantic Coverage*

To verify the comparison performance of Sandiff, we did experiments between different branches of *commercially-available* images of AOSP. The AOSP contains numerous types of files (text, audio, video, symbolic links, binary files, among others) that can be compared semantically. The experiments consist of comparing the following image pairs:

- **Experiment #1:** Analysing two major AOSP with minor revisions: 8.1.0 r64 x 8.1.0 r65.

- **Experiment #2:** Analysing the last revision of AOSP Oreo and initial release of AOSP Pie: 8.1.0 r65 x 9.0.0 r1.

- **Experiment #3:** Analysing the last revision of AOSP Pie and its initial release: 9.0.0 r1 x 9.0.0 r45.

- **Experiment #4:** Analysing two major releases of AOSP Oreo and AOSP 10: 8.1.0 r77 x 10.0.0 r39.

- **Experiment #5:** Analysing two major releases of AOSP Pie and AOSP 10: 9.0.0 r57 x 10.0.0 r39.

These pairs were compared using both semantic (Sandiff) and byte-to-byte (checksum) comparison methods. To demonstrate the robustness of each method, we analysed the files contained in `system.img`, `userdata.img` and `vendor.img` images, which are mounted in the EXT2 file system under a UNIX system. Note that, differently from Sandiff, the byte-to-byte comparison cannot read empty files and symbolic link targets. These files are listed as *errors*, as shown in Table VIII.

Based on the experiments of Table VIII, we can note that Sandiff was able to analyze large software projects like the AOSP. First, the semantic comparison was able to determine

TABLE VIII. OVERALL SUMMARY OF THE IMPACT OF USING SANDIFF IN REAL-WORLD *COMMERCIALLY-AVAILABLE* AOSP BUILDS.

Comparison	Add		Remove		Edit		Type Edit		Equal		Error		Ignored	
	Semantic	Binary	Semantic	Binary	Semantic	Binary	Semantic	Binary	Semantic	Binary	Semantic	Binary	Semantic	Binary
Experiment #1	0	0	0	0	11	12	0	0	2185	2165	0	19	0	0
Experiment #2	13	13	27	27	0	0	3	3	0	0	0	0	0	0
Experiment #3	23	23	18	18	527	606	0	0	1929	1805	0	45	0	0
Experiment #4	179	179	41	41	98	97	5	5	153	154	0	0	0	0
Experiment #5	439	439	240	240	839	844	11	11	785	780	0	0	0	0

* **Add** = file is present on the second input. **Remove** = file is present in the first input. **Edit** = file is present in both inputs, but the comparison returned differences. **Type Edit** = file is present in both inputs, but there were changes in its metadata (e.g., permissions). **Equal** = file is present in both inputs, and the comparison returns an *equal* status. **Error** = file is present in both inputs, but the comparison returns an *error* status. **Ignored** = file is present in both inputs, but is not semantically relevant, so it was ignored.

the file type and compare the file contents and its metadata. In contrast, a byte-to-byte comparison was unable to compare the symbolic link's targets and broken links. Second, the semantic comparison was able to discard irrelevant differences (e.g., the build time in `build.prop`) which are no differences in functionality.

Note that, during *experiment #2*, Sandiff is unable to perform a full analysis between these trees because there were structural changes. For instance, in AOSP Oreo, the `/bin` is a directory containing many files. In contrast, in AOSP Pie, the `/bin` is now a symbolic link to another directory path (that can be another image). As a result, Sandiff detects this case as a *Type Edit* and does not traverse `/bin` since it is only a directory in AOSP Oreo.

The *experiment #3* is similar to *experiment #1*, except that the number of edited files is significantly more extensive since the code has changed due to the different revisions. We notice that *errors* occur in symbolic links, as expected for byte-to-byte comparison. Some files only changed in terms of data, but not in semantic meaning, making this the optimal scenario for Sandiff over the traditional checksum.

Both *experiments #4* and *#5* evaluate at which point the semantic comparison becomes irrelevant, i.e., they exploit Sandiff performance when analyzing significantly different AOSP releases. As expected for these scenarios, the results of both semantic and byte-to-byte comparisons are similar. In summary, the *semantic comparison* is inaccurate when analyzing files that are not recognized by the rules in the current implementation of Sandiff, making the byte-to-byte more appropriate in these cases. Nevertheless, Sandiff is able to support both *semantic* and *non-semantic* tasks, despite run-time performance disadvantages when compared with naive solutions.

B. Scalability

To study the behavior of Sandiff when dealing with multi-threaded AOSP build systems, we performed a scalability evaluation that measures the run-time performance on different (i) execution modes, (ii) number of concurrent *comparison* jobs, and (iii) AOSP builds. In this experiment, we used a workstation-based setup with an Intel Core i7-2600 at 3.40GHz with 16GB of memory, hereafter called *Machine #1*, and a data center server with an Intel Xeon E5-2697 at 2.30GHz with 125GB of memory, hereafter called *Machine #2*.

The *execution modes* are responsible for defining the parallel and recursive operations of Sandiff's *Orchestrator*, as defined in Section III. In summary, it manages the strategies for resource sharing and how comparison results are collected. Below we list the evaluated modes:

- **Walk First:** leverages multi-threading by sequentially analyzing the directories, distributing its files across the comparison jobs. It is the default mode of the Sandiff.
- **Parallel Walk:** performs concurrent directory analysis up to the number of comparison jobs. It is the recommended mode for analyzing directories with a large number of files.
- **Mixed:** iterates in both files and directories.
- **Slice:** lists all files before processing, then distributes them in similar batches across the maximum number of comparison jobs.

To minimize the variance between runs, we repeated each experiment four times, as shown in Figures 7 and 8. Note that, despite different scenarios, Sandiff achieved its best run-time performance when running with four parallel comparison jobs. Due to AOSP nature, Sandiff cannot successfully parallelize the jobs since the tasks are interdependent. In general, *Walk First*, *Parallel Walk*, and *Mixed* modes tend to attain similar scalability.

To cope with the variance, we repeated each experiment four times, as shown in Figures 7 and 8. Note that, despite different scenarios, run-time performance increases quickly as the number of jobs grows. Due to the small amount of data in pure AOSP images - not more than 1900 files and 800MB - the orchestration process among multiple jobs creates an overhead that is not compensated by the parallelization after four parallel jobs. This limitation is overcome when larger images are used. To illustrate the scenario where more data is compared, we run a comparison between two commercially-available builds based on AOSP having 4847 files and a total size of 4GB. As can be seen in Figure 9, parallel comparison stands up when a higher amount of data is compared. Execution time decreases as the number of jobs gets closer to the number of physical cores available on the machine.

In general, *Walk First*, *Parallel Walk*, and *Mixed* modes tend to attain similar scalability, but *Slice* mode provided better performance, relatively and absolutely, when the number of jobs coincides with the number of available cores of the machine. This occurs due to the decreased number of context changes provided by the Slice mode combined with the maximum usage of available cores.

VI. CONCLUSION

In this paper, we presented *Build Comparator*, an integrated tool for supporting software releases and DevOps operations

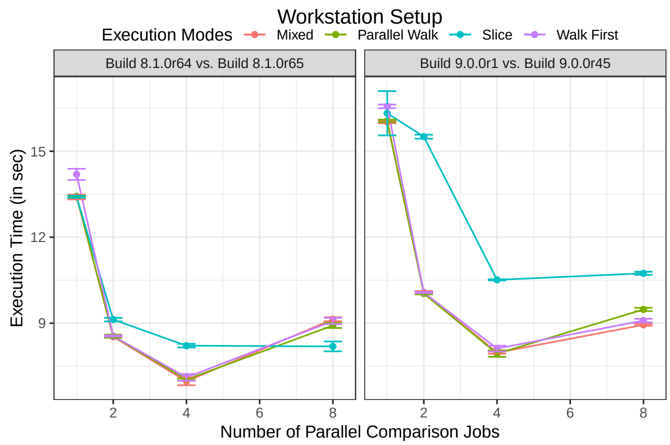


Figure 7. Scalability performance when analysing AOSP builds on workstation-based setups.

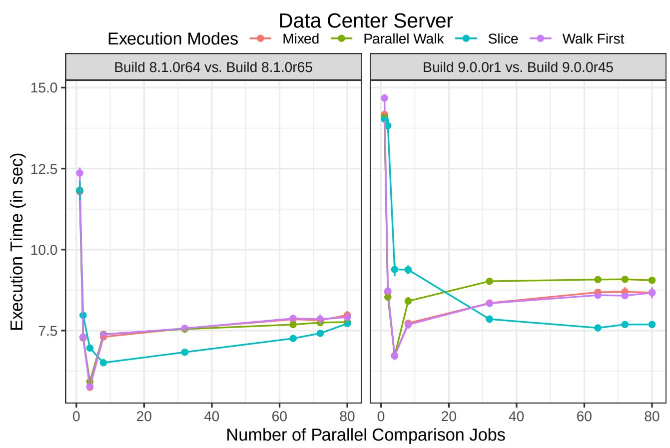


Figure 8. Scalability performance when analysing AOSP builds on data center servers, i.e., dedicated environments for experimenting.

on Android development pipelines by leveraging Sandiff, a semantic comparator designed to facilitate continuous testing of large software projects, specifically those related to AOSP. To the best of our knowledge, Sandiff is the first to allow correlation of test routines of the official Android Test Suite (VTS) with semantic modifications in intermediate files of AOSP building process. When used to skip time-consuming *test cases* or to mount a list of priority tests (*fail-fast*), Sandiff can lead to higher productivity of mobile developers. We showed that semantic comparison is more robust to analyze large projects than binary comparison since the latter cannot discard irrelevant modifications to the target software's output or execution. As we refine the semantic comparators of Sandiff, more AOSP specific rules will apply, and consequently, more items can be classified as "Equal" in Sandiff's comparison reports.

With *Build Comparator*, we presented and analyzed an architecture that enables the integration of *semantic comparison* with systems that are commonly used in the development of AOSP software, exploiting *real-world* use cases. In the context of making Sandiff domain agnostic, another avenue for future work is to explore machine learning techniques

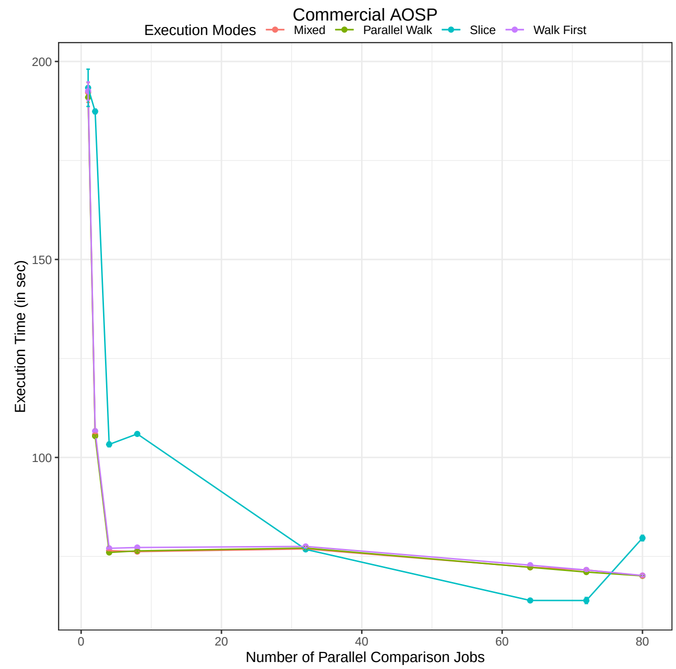


Figure 9. Scalability performance when analysing a large commercial AOSP build on a data center server.

to detect how tests are related to different files and formats. We also plan to extend *Build Comparator*'s reporting features by proposing visualizations that highlight relevant semantic differences between pairs of files and integrate Sandiff to the official Android Test Suite (VTS) to validate our intermediate results.

ACKNOWLEDGMENTS

We thank both Rafael Melo da Silva and Nick Diego Yamane Pinto for their valuable help during the development of the project. This work was partially supported by Samsung Eletrônica da Amazônia Ltda., under the auspice of the informatics law no 8.387/91.

REFERENCES

[1] C. E. D. S. Aguiar, J. I. B. V. Filho, A. O. P. Junior, R. J. B. Fernandes, and C. A. D. L. Pahins, "Sandiff: Semantic file comparator for continuous testing of android builds," VALID 2019 : The Eleventh International Conference on Advances in System Testing and Validation Lifecycle, Nov. 2019, pp. 51–55.

[2] Araxis Ltd. Araxis: Software. [Online]. Available: https://www.araxis.com/ [retrieved: November, 2020]

[3] Free Software Foundation, Inc. Diffutils. [Online]. Available: https://www.gnu.org/software/diffutils/ [retrieved: October, 2020]

[4] C. J. Madsen. Vbindiff - visual binary diff. [Online]. Available: https://www.cjmweb.net/ [retrieved: October, 2020]

[5] Y. Wang, D. J. DeWitt, and J. Cai, "X-diff: an effective change detection algorithm for xml documents," in Proceedings 19th International Conference on Data Engineering, March 2003, pp. 519–530.

[6] M. Pawlik and N. Augsten, "Efficient computation of the tree edit distance," ACM Transactions on Database Systems, vol. 40, 2015, pp. 3:1–3:40.

[7] F. Foscarin, F. Jacquemard, and R. Fournier-S'niehotta, "A diff procedure for music score files," in 6th International Conference on Digital Libraries for Musicology, ser. DLfM '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 58–64. [Online]. Available: https://doi.org/10.1145/3358664.3358671

[8] G. Barabucci, “Diffi: Diff improved; a preview,” in Proceedings of the ACM Symposium on Document Engineering 2018, ser. DocEng ’18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3209280.3229084

[9] I. Sorokin, “Comparing files using structural entropy,” Journal in Computer Virology, vol. 7, 2011. [Online]. Available: https://doi.org/10.1007/s11416-011-0153-9

[10] R. M. A. Mohammad and M. Alqahtani, “A comparison of machine learning techniques for file system forensics analysis,” Journal of Information Security and Applications, vol. 46, 2019, pp. 53 – 61. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214212618307579

[11] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, “A comparative study of message digest 5(MD5) and SHA256 algorithm,” Journal of Physics: Conference Series, vol. 978, 2018, pp. 1–6.

[12] G. Leurent and T. Peyrin, “From collisions to chosen-prefix collisions application to full sha-1,” in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2019, pp. 527–555.

[13] R. Martino and A. Cilardo, “Sha-2 acceleration meeting the needs of emerging applications: A comparative survey,” IEEE Access, vol. 8, 2020, pp. 28 415–28 436.

[14] D. M. A. Cortez, A. M. Sison, and R. P. Medina, “Cryptographic randomness test of the modified hashing function of sha256 to address length extension attack,” in Proceedings of the 2020 8th International Conference on Communications and Broadband Networking, 2020, pp. 24–28.

[15] P. M. Simanullang, S. Sinurat, and I. Saputra, “Analisa metode sha384 untuk mendeteksi orisinalitas citra digital,” KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer), vol. 3, no. 1, 2019.

[16] A. Jose and K. Subramaniam, “Dna based sha512-ecc cryptography and cm-csa based steganography for data security,” Materials Today: Proceedings, 2020.

[17] S. Long, “A comparative analysis of the application of hashing encryption algorithms for MD5, SHA-1, and SHA-512,” Journal of Physics: Conference Series, vol. 1314, Oct. 2019, p. 012210.

[18] G. Shaheen, “A robust review of sha: Featuring coherent characteristics,” International Journal of Computer Science and Mobile Computing, vol. 9, 2020, p. 111–116.

[19] FreeType Project. Freetype. [Online]. Available: https://www.freetype.org/freetype2/ [retrieved: November, 2020]

[20] Apache Software Foundation. C xml parser. [Online]. Available: https://xerces.apache.org/xerces-c/ [retrieved: October, 2020]

[21] PMEase. Quickbuild. [Online]. Available: https://www.pmease.com/ [retrieved: October, 2020]

International Journal on Advances in Software, vol 13 no 3 & 4, year 2020, http://www.iariajournals.org/software/

On the Realization of Meta-Circular Code Generation and Two-Sided Collaborative Metaprogramming

Herwig Mannaert
Normalized Systems Institute
University of Antwerp
Antwerp, Belgium
herwig.mannaert@uantwerp.be

Koen De Cock
Research & Development
NSX bv
Niel, Belgium
koen.de.cock@nsx.normalizedsystems.org

Peter Uhnák
Research & Development
NSX bv
Niel, Belgium
peter.uhnak@nsx.normalizedsystems.org

Jan Verelst
Normalized Systems Institute
University of Antwerp
Antwerp, Belgium
jan.verelst@uantwerp.be

***Abstract*—The automated generation of source code is a widely adopted technique to improve the productivity of computer programming. Normalized Systems Theory (NST) aims to create software systems exhibiting a proven degree of evolvability. A software implementation exists to create skeletons of Normalized Systems (NS) applications, based on automatic code generation. This paper describes how the NS model representation, and the corresponding code generation, has been made meta-circular, and presents its detailed architecture. It is argued that this feature may be crucial to improve the productivity of metaprogramming, as it enables scalable collaboration based on two-sided interfaces. Some preliminary results from applying this approach in practice are presented and discussed.**

Index Terms*—*Evolvability, meta-circularity, normalized systems, automatic programming; case study

I. Introduction

This paper extends a previous paper that was originally presented at the Fourteenth International Conference on Software Engineering Advances (ICSEA) 2019 [1].

Increasing the productivity in computer programming has been an important and long-term goal of computer science. Though many different approaches have been proposed, discussed, and debated, two of the most fundamental approaches toward this goal are arguably automated code generation and homoiconic programming. Increasing the evolvability of Information Systems (IS) on the other hand, is crucial for the productivity during the maintenance of information systems. Although it is even considered as an important attribute determining the survival chances of organizations, it has not yet received much attention within the IS research area [2]. Normalized Systems Theory (NST) was proposed to provide an ex-ante proven approach to build evolvable software by leveraging concepts from systems theory and statistical thermodynamics. In this paper, we present an integrated approach that combines both Normalized Systems Theory to provide improved evolvability, and automated code generation and homoiconic programming to offer increased productivity. We also argue that this combined approach can enable entirely new levels of productivity and scalable collaboration.

The remainder of this paper is structured as follows. In Section II, we briefly discuss two fundamental approaches to increase the productivity in computer programming: automatic and homoiconic programming. In Section III, we give an overview of NST as a theoretical basis to obtain higher levels of evolvability in information systems, and discuss the NST code generation or expansion. Section IV presents the realization of the meta-circular metaprogramming architecture, and details the declarative control structure. Section V elaborates on the possibilities that the two-sided interfaces of the metaprogramming architecture offer for scalable collaboration. Finally, we report and discuss some results in Section VI, and present our conclusions in Section VII.

II. Automatic and Homoiconic Programming

A. Automatic or Metaprogramming

The automatic generation of code is nearly as old as coding or software programming itself. One often makes a distinction between *code generation*, the mechanism where a compiler generates executable code from a traditional high-level programming language, and *automatic programming*, the act of automatically generating source code from a model or template. In fact, one could argue that both mechanisms are quite similar, as David Parnas already concluded in 1985 that "automatic programming has always been a euphemism for programming in a higher-level language than was then available to the programmer" [3]. In general, automatic programming performs a transformation from domain and/or intermediate models to programming code.

Another term used to designate automatic programming is *generative programming*, aiming to write programs ”to manufacture software components in an automated way” [4], in the same way as automation in the industrial revolution has improved the production of traditional artifacts. As this basically corresponds to an activity at the meta-level, i.e., writing software programs that write software programs, this is also referred to as *metaprogramming*. Essentially, the goal of automatic programming is and has always been to improve programmer productivity.

Software development methodologies such as *Model-Driven Engineering (MDE)* and *Model-Driven Architecture (MDA)*, focusing on creating and exploiting conceptual domain models and ontologies, are also closely related to automatic programming. In order to come to full fruition, these methodologies require the availability of tools for the automatic generation of source code. Currently, these model-driven code generation tools are often referred to as *Low-Code Development Platforms (LCDP)*, i.e., software that provides an environment for programmers to create application software through graphical user interfaces and configuration instead of traditional computer programming. As before, the goal remains to increase the productivity of computer programming.

The field is still evolving while facing various challenges and criticisms. Some question whether low-code development platforms are suitable for large-scale and mission-critical enterprise applications [5], while others even question whether these platforms actually make development cheaper or easier [6]. Moreover, defining an intermediate representation or reusing *Domain Specific Languages (DSLs)* is still a subject of research today. We mention the contributions of Wortmann [7], presenting a novel conceptual approach for the systematic reuse of *Domain Specific Languages*, Gusarov et al. [8], proposing an intermediate representation to be used for code generation, and Frank [9], pleading for multi-level modeling. Hutchinson et al. elaborate on the importance of organizational factors for code generation adoption [10], and suggest that the benefit of model-driven development has to be found in a more holistic approach to software architecture [11]. We have argued in our previous work that some fundamental issues need to be addressed, like the increasing complexity due to changes during maintenance, and have proposed to combine automatic programming with the evolvability approach of *Normalized Systems Theory (NST)* [12].

B. Homoiconicity or Meta-Circularity

Another technique in computer science aimed at the increase of the abstraction level of computer programming, thereby aiming to improve the productivity, is homoiconicity. A language is homoiconic if a program written in it can be manipulated as data using the language, and thus the program’s internal representation can be inferred just by reading the program itself. As the primary representation of programs is also a data structure in a primitive type of the language itself, reflection in the language depends on a single, homogeneous structure instead of several different structures. It is this language feature that can make it much easier to understand how to manipulate the code, which is an essential part of metaprogramming. The best known example of an homoiconic programming language is Lisp, but all Von Neumann architecture systems can implicitly be described as homoiconic. An early and influential paper describing the design of the homoiconic language TRAC [13], traces the fundamental concepts back to an even earlier paper from McIlroy [14].

Related to homoiconicity is the concept of a *meta-circular evaluator (MCE)* or *meta-circular interpreter (MCI)*, a term that was first coined by Reynolds [15]. Such a meta-circular interpreter, most prominent in the context of Lisp as well, is an interpreter which defines each feature of the interpreted language using a similar facility of the interpreter’s host language. The term meta-circular clearly expresses that there is a connection or feedback loop between the activity at the meta-level, the internal model of the language, and the actual activity, writing models in the language.

There is a widespread belief that this kind of properties increase the abstraction level and therefore the productivity of programming. We will argue that this is even more relevant for automatic programming, as the metaprogramming code, i.e., the programming code generating the code, is often complex and therefore hard to maintain. Moreover, the potential of meta-circularity with respect to productivity can be seen in the context of other technologies. For instance, a transistor is a switch that can be switched by another transistor. Therefore, when a smaller and faster transistor/switch is developed, there is no need to develop a new version of the switching device, as such a new version of this device, i.e., the transistor itself, is already there, and smaller and faster as well. Such a shortcut of the design cycle can clearly foster rapid progress.

III. Normalized Systems Theory and Expansion

In this section, we discuss the code generation or *expansion* based on Normalized Systems Theory, attempting to address some fundamental issues that automatic programming is facing today: the lack of evolvability in information systems, and the increasing complexity due to changes.

A. Evolvability and Normalized Systems

The evolvability of information systems (IS) is considered as an important attribute determining the survival chances of organizations, although it has not yet received much attention within the IS research area [2]. Normalized Systems Theory (NST), applying the concept of *stability* from systems theory to the design cycle of information systems, was proposed to provide an ex-ante proven approach to build evolvable software [12], [16], [17]. Systems theoretic stability is an essential property of systems, and means that a bounded input should result in a bounded output. In the context of information systems development and evolution, this implies that a bounded set of changes should result in a bounded set of impacts to the software. Put differently, it is demanded that the impact of changes to an information system should not be dependent on the size of the system to which they are applied, but only on the size of the changes to be performed. Changes rippling through and causing an impact dependent on the size of the system are called *combinatorial effects*, and are considered to be a major factor limiting the evolvability of information systems. The theory prescribes a set of theorems, and formally proves that any violation of any of the following *theorems* will result in combinatorial effects, thereby hampering evolvability [12], [16], [17]:

- *Separation of Concerns:* Every concern, defined as an independent change driver, should be separated in its own class or module.
- *Action Version Transparency:* Every computing action should be encapsulated to shield other modules from internal implementation changes.
- *Data Version Transparency:* Every data structure should be encapsulated to shield modules passing this data from internal data changes.
- *Separation of States:* Every result state from a computing action should be stored, to shield other modules from having to deal with new types of implementation errors.

The application of the theorems in practice has shown to result in very fine-grained modular structures within a software application. In particular, the so-called cross-cutting concerns, i.e., concerns cutting across the functional structure, need to be separated as well. This is schematically represented for three domain entities ('Order', 'Invoice', and 'Payment'), and three cross-cutting concerns ('Persistency', 'Access Control', and 'Remote Access') in Figure 1. Though the actual implementation of the cross-cutting concern is in general provided by an external framework (represented by the colored planes), the code connecting to that framework (represented by the small colored disks), is considered to be a change driver, and needs to be separated and properly encapsulated.

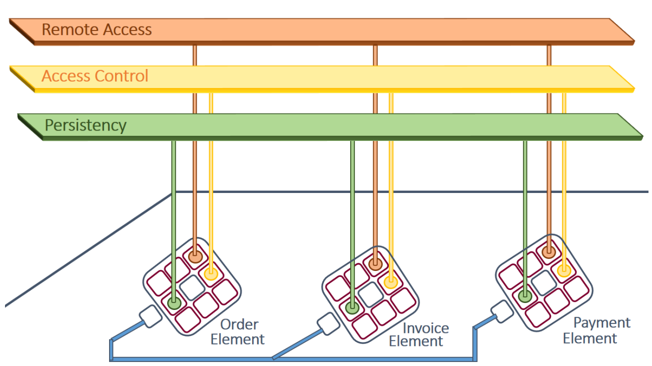


Fig. 1. Representation of domain entities connecting to external frameworks.

As every domain entity needs to connect to various additional external frameworks providing cross-cutting concerns, e.g., 'Transaction' or 'REST Service', these entities need to be implemented by a *set of classes*. Such a set of classes, schematically represented in Figure 2 for the domain entity 'Invoice', is called an *Element* in NST.

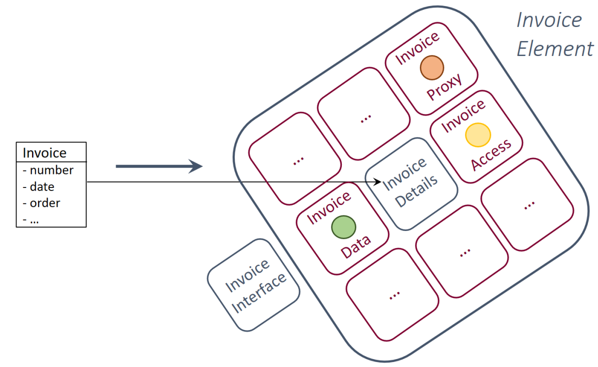


Fig. 2. Representation of an *element* for the domain entity 'Invoice'.

Such structures are, in general, difficult to achieve through manual programming. Therefore, the theory also proposes a set of patterns to generate significant parts of software systems which comply with these theorems. More specifically, NST defines five types of *elements* to provide the main functionality for information systems, and proposes five detailed design patterns to implement these element structures [17] [12]:

- *Data element* to represent a data or domain entity.
- *Action element* to implement a computing action or task.
- *Workflow element* to orchestrate a flow or state machine.
- *Connector element* to provide a user or service interface.
- *Trigger element* to perform a task or flow periodically.

The implementation or instantiation of the element structures results in a codebase with a highly recurring structure. Such a recurring structure is desirable as it *increases the consistency*, and *reduces the complexity* of the codebase. However, this recurring structure will have to be adapted over time based on new insights, the discovery of flaws, and/or changes in underlying technologies or frameworks. These structural changes may need to be applied in a retroactive way, but the efforts increase with the frequency of these adaptations. For instance, if one decides to adapt or refactor the element structures in a growing system in a retroactive way every time K additional elements have been created, the total amount of refactored element structures when the system reaches N different elements, will be equal to:

$$K + 2K + ... + N = \sum_{i=1}^{N/K} i.K = \frac{N(N+K)}{2K} \qquad (1)$$

Therefore, the element structures of NST software are generated *and regenerated* in a —rather straightforward— automated way. First, a model of the considered universe of discussion is defined in terms of a set of data, task and workflow elements. Next, code generation or automated programming is used to generate parametrized copies of the general element design patterns into boiler plate source code. Due to the simple and deterministic nature of this code generation mechanism, i.e., instantiating parametrized copies, it is referred to as *NS expansion* and the generators creating the individual coding artifacts are called *NS expanders*. This generated code can, in general, be complemented with custom code or *craftings* to add non-standard functionality that is not provided by the expanders themselves, at specific places within the boiler plate code marked by *anchors*. This custom code can be automatically *harvested* from within the anchors, and *re-injected* when the recurring element structures are regenerated.

B. Expansion and Variability Dimensions

In applications generated by a Normalized Systems (NS) expansion process, schematically represented in Figure 3 around the symbolic blue icon, we identify four variability dimensions. As discussed in [18] [19], the combination of these dimensions compose an actual NS application codebase, represented in the lower right of Figure 3, and therefore determine how such an application can evolve through time, i.e., how software created in this way exhibits evolvability.

First, as represented at the upper left of the figure, one should specify or select the *models* or *mirrors* he or she wants to expand. Such a model is technology agnostic (i.e., defined without any reference to a particular technology that should be used) and represented by standard modeling techniques, such as ERD's for data elements and BPMN's for task and flow elements. Such a model can have multiple versions throughout time (e.g., being updated or complemented) or concurrently (e.g., choosing between a more extensive or summarized version). As a consequence, the chosen model represents a first dimension of variability or evolvability.

Second, as represented on top of the blue icon in the figure, one should provide the parametrized *coding templates* for the various classes of the elements according to a specific element structure as represented in Figure 2. The *expanders* will generate (boiler plate) source code by instantiating the various class templates or *skeletons* of the element structures, i.e., the design patterns, taking the specifications of the model as *parameters*. For instance, for a data element 'Invoice', a set of java classes *InvoiceDetails*, *InvoiceProxy*, *InvoiceData*, *InvoiceAccess*, etcetera will be generated. This code can be considered boiler plate code as it provides a set of standard functionalities for each of the elements within the model, though they have evolved over time to provide features like standard finders, master-detail (waterfall) screens, certain display options, document upload/download functionality, child relations, etcetera. The expanders and corresponding template skeletons evolve over time as improvements are made and bugs are fixed, and as additional features (e.g., creation of a status graph) are provided. Given the fact that the application model is completely technology agnostic, and that it can be used for any version of the expanders, these bug fixes and additional features become available for all versions of all application models: only a *re-expansion* or *"rejuvenation"* is required. As a consequence, the expanders or template skeletons represent a second dimension of variability or evolvability.

Third, as represented in the upper right of the figure, one should specify *infrastructural options* to select a number of frameworks or *utilities* to take care of several generic or so-called cross-cutting concerns. These options consist of global options (e.g., determining the build automation framework), presentation settings (determining the graphical user interface frameworks), business logic settings (determining the database and transaction framework to be used) and technical infrastructure (e.g., selecting versions for access control or persistency frameworks). This means that, given a chosen application model version and expander version, different variants of boiler plate code can be generated, depending on the choices regarding the infrastructural options. As a consequence, the settings and utility frameworks represent a third dimension of variability or evolvability.

Fourth, as represented in the lower left of the figure, "custom code" or *craftings* can be added to the generated source code. These craftings enrich, i.e., are put upon, the earlier generated boiler plate code. They can be *harvested* into a separate repository before regenerating the software application, after which they can *re-injected*. The craftings include extensions, i.e., additional classes added to the generated code base, as well as insertions, i.e., additional lines of code added between the foreseen anchors within the code. Craftings can have multiple versions throughout time (e.g., being updated or complemented), or concurrently (e.g., choosing between a more advanced or simplified version). These craftings should contain as little technology specific statements within their source code as possible (apart from the chosen background technology, i.e., the programming language). Indeed, craftings referring to (for instance) a specific UI framework will only be reusable as long as this particular UI framework is selected for the generation of the application. In contrast, craftings performing certain validations, but not containing any specific references to the transaction framework, e.g., *Enterprise Java Beans (EJB)*, can simply be reused when applying other versions or choices regarding such a framework. As a consequence, the custom code or craftings represent a fourth dimension of variability or evolvability.

In summary, each part in Figure 3 is a variability dimension in an NST software development context. It is clear that talking about *the* "version" of an NST application, as is traditionally done for software systems, becomes more refined in such a context. Indeed, the eventual software application codebase (the lower right side of the figure) is the result of a specific version of an application model, expander version, infrastructural options, and a set of craftings [19]. Put differently, with M, E, I and C referring to the number of available application model versions, the number of expander versions, the number of infrastructural option combinations, and the number of crafting sets respectively, the total set of possible versions V of a particular NST application becomes equal to:

$$V = M \times E \times I \times C \tag{2}$$

Whereas the specific values of M and C are different for every single application, the values of E and I are dependent on the current state of the expanders. Remark that the number of infrastructural option combinations (I) is equally a product:

$$I = G \times P \times B \times T \tag{3}$$

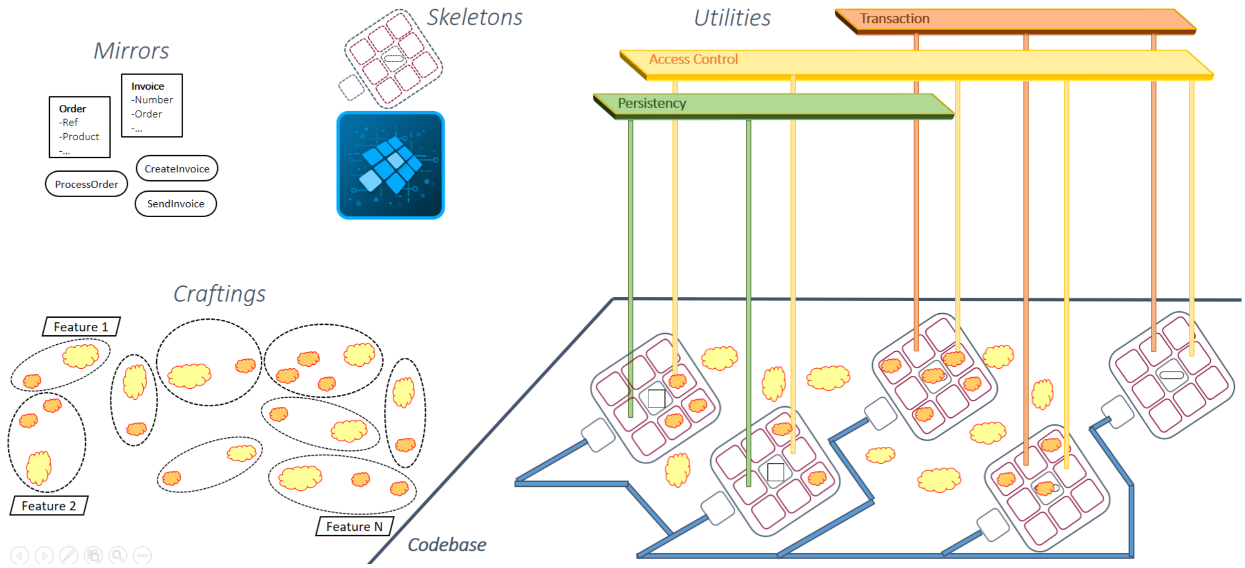


Fig. 3. A graphical representation of four variability dimensions within a Normalized Systems application codebase.

In this case, G represents the number of available global option settings, P the number of available presentation settings, B the number of available business logic settings, and T the number of available technical infrastructure settings. This general idea in terms of combinatorics corresponds to the overall goal of NST: enabling evolvability and variability by *leveraging the law of exponential variation gains* by means of the thorough decoupling of the various concerns, and the facilitation of their recombination potential [12].

IV. Meta-Circular Expansion Software

In this section, we present the meta-circular architecture of the NST expansion software, i.e., the automatic programming code that is also able to regenerate itself as well. Both the sequential phases toward achieving this meta-circular code (re)generation architecture, and the declarative control structure of the expansion software, are described.

A. Toward Meta-Circular Expansion

1) Phase 1: Standard Code Generation: The original architecture of the Normalized Systems expansion or code generation software is schematically represented in Figure 4. On the right side of the figure, the generated source code is represented in blue, corresponding to a traditional multi-tier web application. Based on a *Java Enterprise Edition (JEE)* stack [17] [19], the generated source code classes are divided over so-called layers, such as the logic, the control, and the view layer. On the left side, we distinguish the internal structure of the expanders or the code generators, represented in red. This corresponds to a very straightforward implementation of code generators, consisting of:

- *model files* containing the model parameters.
- *reader classes* to read the model files.
- *model classes* to represent the model parameters.
- *control classes* selecting and invoking the different expander classes based on the parameters.
- *expander classes* instantiating the source templates, using the *String Template (ST)* library, and feeding the model parameters to the source templates.
- *source templates* containing the parametrized code.

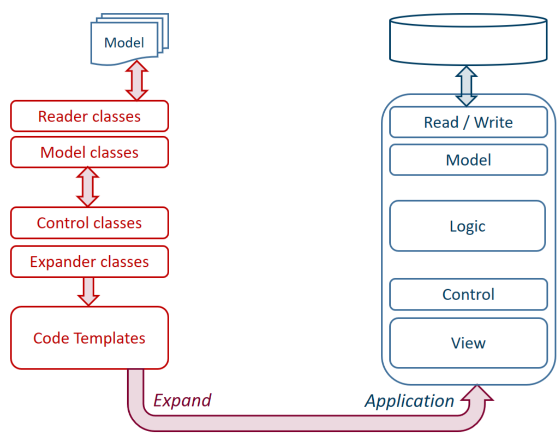


Fig. 4. Representation of a basic code generator structure.

2) Phase 2: Generating a Meta-Application: Essentially, code generation models or meta-models — and even all collections of configuration parameters — consist of various data entities with attributes and relationships. As the Normalized Systems element definitions are quite straightforward [17] [19], the same is valid for its meta-models. Moreover, one of the Normalized Systems elements, i.e., the data element, is basically a data entity with attributes. This means that the NS meta-models, being data entities with attributes, can be expressed as regular models. For instance, in the same way 'Invoice' and 'Order' can be specified as NS data elements with attributes and relationships in an information system model, the NS 'data element' and 'task element' of the NS meta-model can be defined as NS data elements with attributes and relationships, just like any other NS model.

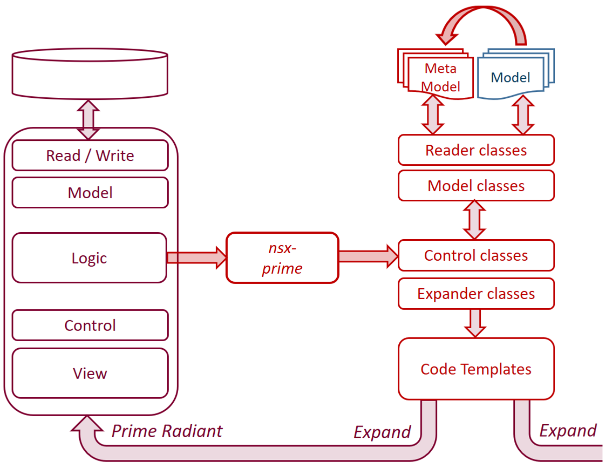


Fig. 5. Expansion of a meta-application to define meta-models.

As the NS models can be considered a higher-level language according to Parnas [3], the single structure of its model data and meta-model language means that the NS model language is in fact homoiconic in the sense of [14]. This also enables us to expand or generate a meta-application, represented on the left side of Figure 5 in dark red. This NS meta-application, called the *Prime Radiant*, is a multi-tier web application, providing the functionality to enter, view, modify, and retrieve the various NS models. As the underlying meta-model is just another NS model, the Prime Radiant also provides the possibility to view and manipulate its own internal model. Therefore, by analogy with the meta-circular evaluator of Reynolds [15], the Prime Radiant can be considered to be a *meta-circular application*.

For obvious reasons, the generated reader and model classes (part of the Prime Radiant on the left side of Figure 5) slightly differ from the reader and model classes that were originally created during the conception of the expansion or code generation software (on the right side of Figure 5). This means that in order to trigger and control the actual expansion classes to generate the source code, an integration software module needed to be developed, represented in the middle of Figure 5 as *nsx-prime*. Though the Prime Radiant meta-application is auto-generated, and can therefore be regenerated or rejuvenated as any NS application, this *nsx-prime* integration module needed to be maintained manually.

3) Phase 3: Closing the Expander Meta-Circle: Though the original reader and model classes of the expander software differed from the generated reader and writer classes, there is no reason that they should remain separate. It was therefore decided to perform a rewrite of the control and expander classes of the expander software (on the right side of Figure 5), to allow for an easier integration with the auto-generated reader and model classes (on the left side of Figure 5). Enabling such a near-seamless integration would not only eliminate the need for the reader and model classes of the expander software, it would also reduce the complexity of the *nsx-prime* integration component to a significant extent.

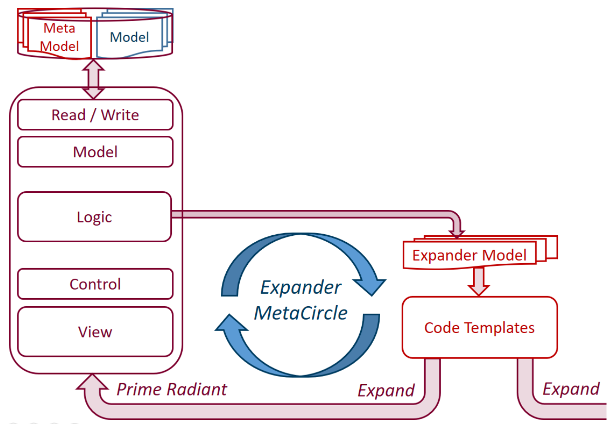


Fig. 6. Closing the meta-circle for expanders and meta-application.

Originally, the refactoring was only aimed at the elimination of the reader and control classes of the expander software. During the refactoring however, it became clear that the control and expander classes of the expander software implementation could be eliminated as well. Indeed, by adopting a declarative structure to define the expander templates and to specify the relevant model parameters, both the control classes (selecting and invoking the expander classes) and the expander classes (instantiating and feeding the parameters to the source templates) were no longer necessary. Moreover, as schematically represented in Figure 6, the refactoring also eliminated the need for the *nsx-prime* integration module. As extensions to the meta-model no longer require additional coding in the various expander software classes (e.g., reader, model, control, and expander classes), nor to the *nsx-prime* integration module, one can say that the *expander development meta-circle* has been closed. This is symbolically visualized in Figure 6. Indeed, expander templates can be introduced by simply defining them, and extensions to the NS meta-model become automatically available after re-running the expansion or code generation on this meta-model.

B. Declarative Expansion Control

The expansion process of an NS element is schematically represented in Figure 7. The internal structure of an NS element (data, task, flow, connector, and trigger element) is based on a detailed design pattern [16] [17] [12], implemented through a set of source code templates. We call this set of coding templates the *Element Template*, represented in dashed lines on the left side of Figure 7. During the actual expansion or code generation, for every instance of an NS element, e.g., a data element 'Invoice', the set of source code templates is instantiated, steered by the parameters of the model. This results in the actual element, e.g., *Invoice Element*, which is represented in solid lines and corresponds to Figure 2.

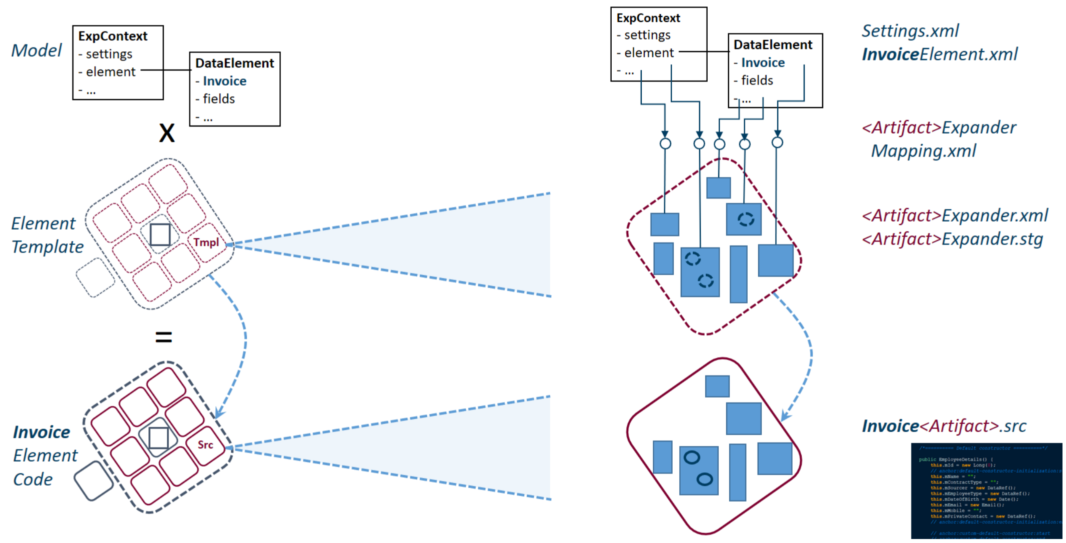


Fig. 7. Expansion of an *Invoice* element, zooming in on a single artifact.

1) Declarative Representation of Expanders: On the right side of Figure 7, we zoom in on the expansion of an individual source code template into a source code artifact, e.g., a class, applying and inserting the parameters of the model. We refer to this unit of code generation as an individual *expander*, and mention that the NS code generation environment for web-based information systems currently consists of 182 individual expanders. Every individual expander is declared in an XML document. An example of such an expander declaration, defining various properties, is shown below.

```
<expander name="DataExpander"
  xmlns="http://normalizedsystems.org/expander">
  <packageName>expander.jpa.dataElement</packageName>
  <layerType name="DATA_LAYER"/>
  <technology name="JPA"/>
  <sourceType name="SRC"/>
  <elementTypeName>DataElement</elementTypeName>
  <artifact>$dataElement.name$Data.java</artifact>
  <artifactPath>$componentRoot$/$artifactSubFolders$/
    $dataElement.packageName</artifactPath>
  <isApplicable>true</isApplicable>
  <active value="true"/>
  <anchors/>
</expander>
```

In this declaration, we find the following information.

- The identification of the expander, name and package name, which also identifies in an unambiguous way the source code template.
- Some technical information, including the tier or layer of the target artifact in the application, the technology it depends on, and the source type.
- The name and the complete path in the source tree of the artifact that will be generated, and the type of NS element that it belongs to.
- Some control information, stating the model-based condition to decide whether the expander gets invoked.
- Some information on the anchors delineating sections of custom code that can be harvested and re-injected.

2) Declarative Mapping of Parameters: The instantiation of an individual source code template for an individual instance of an NS element, is schematically represented on the right side of Figure 7. It can be considered as a transformation that combines the parameters of the model with the source code template, and results in a source code artifact. We therefore distinguish three types of documents or artifacts.

- The model parameters, represented at the top of Figure 7, consist of the attributes of the element specification, e.g., the data element 'Invoice' with its fields or attributes, and the options and technology settings. All these parameters are available through the auto-generated model classes, e.g., *InvoiceDetails*, and may either originate from the Prime Radiant database, or from XML files.
- An individual source code template, having a unique name that corresponds to the one of the expander definition as presented above. Such a template, represented in the middle of Figure 7, contains various insertions of parameter values, and parameter-based conditions on the value and/or presence of specific parts of the source code.
- An instantiated source file or artifact, represented at the bottom of Figure 7, where the various values and conditions in the source code template have been resolved.

An important design feature is related to the mapping of the parameters from the model to the parameters that appear in the source code templates, that are directly guiding the code instantiation. In order to provide loose coupling between these two levels of parameters, and to ensure a simple and straightforward relationship, it was decided to implement this mapping in a declarative *ExpanderMapping* XML file. As the entire NS model is made available as a graph of model classes, the parameters in the templates can be evaluated from the NS model using *Object-Graph Navigation Language (OGNL)* expressions. These expressions, e.g., *Invoice.number*, are declared in the XML mapping file of the expander.

V. Two-Sided Scalable Collaboration

In this section, we explain how the meta-circular architecture of the metaprogramming software enables an open and scalable collaboration based on its two-sided interfaces.

A. The Need for Meta-Level Interfaces

The main purpose of an *Application Programming Interface (API)* is to enable widespread and scalable collaboration in software development. It allows for a structured collaboration between developers implementing the interface on one side, and developers using or invoking the programming interface on the other side. The collaboration is possible both within companies and across companies, and in open source communities. The use of such an API has contributed significantly to the rich application offering that we have, including desktop applications and mobile apps, and to the abundant hardware support that we enjoy, providing drivers for a multitude of peripheral devices on a variety of operating systems.

In order to enable scalable collaboration in metaprogramming, we should define meta-level programming interfaces. However, defining meta-level interfaces is still a subject of research today. We have mentioned for instance research papers of Wortmann [7], presenting a novel approach for the systematic reuse of *Domain Specific Languages (DSLs)*, and Gusarov et al. [8], proposing an intermediate representation to be used for code generation. We believe that the complex architectures of metaprogramming environments, exhibiting high degrees of coupling as represented on the left side of Figure 8 (which is similar to the representation in Figure 4), make it nearly impossible to define clear meta-level interfaces.

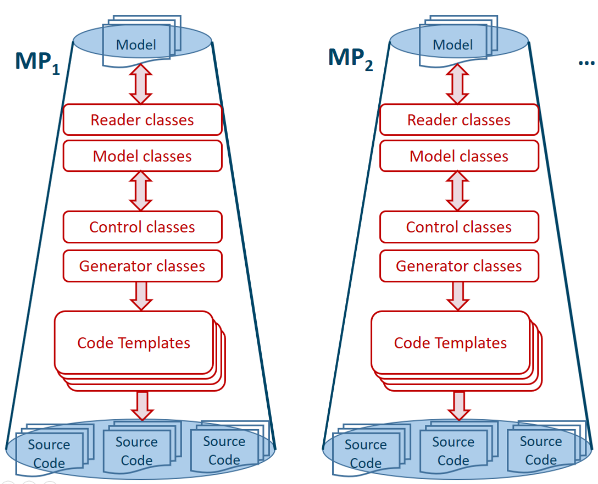


Fig. 8. Representation of various metaprogramming silos.

What all implementations of automatic programming or metaprogramming have in common, is that they perform a transformation from domain models and/or intermediate models to code generators and programming code. This implies that (programming) interfaces need to be defined at both ends of the transformation, allowing at the same time to define or extend domain models, and to implement or replace code generators or templates. We argue that the presented meta-circular architecture enables the definition of the interfaces at both ends of the transformation, as it allows to integrate —or at least accommodate— ever more extensions and alternatives at the two sides of the interface, without entailing the non-scalable burden of adapting the metaprogramming code.

B. Separating Model and Code Interfaces

Similar to the representation in Figure 4, the left hand side of Figure 8 represents a basic code generator or metaprogramming architecture. However, if we consider another metaprogramming environment (for instance on the right side of Figure 8), we will almost certainly encounter a duplication of such an architecture. This will in general result in what we could describe as *metaprogramming silos*, entailing several significant drawbacks. First, it is hard to collaborate between the different metaprogramming silos, as both the nature of the models and the code generators will be different. Second, contributing to the metaprogramming environment will require programmers to learn the internal structure of the model and control classes in the metaprogramming code. As metaprogamming code is intrinsically abstract, this is in general not a trivial task. And third, as contributions of individual programmers will be spread out across the models, readers, control classes, and actual coding templates, it will be a challenge to maintain a consistent decoupling between these different concerns.

The meta-circular environment presented in Section IV-A and Section IV-B addresses these drawbacks. The architecture establishes a clear decoupling between the models and the code generation templates, and removes the need for contributors to get acquainted with the internal structure of the metaprogramming environment. It could allow developers to collaborate in a scalable way at both sides of the metaprogramming interfaces. And, as schematically represented in Figure 9, the clear decoupling of this *horizontal integration* architecture could bring the same kind of variability gains as described in Equations (2) and (3). Indeed, in such a decoupled environment, it should be possible to combine every version or variant of the model with every version or variant of the coding templates to generate a codebase. This implies:

$$N \text{ models } + M \text{ templates } \implies N \times M \text{ codebases} \quad (4)$$

The use of a metaprogramming environment with a clear decoupling between models and code (templates) could also entail significant productivity gains for regular software development projects. Consider for instance Figure 10, representing in a schematic way several collaborative projects, e.g., open source projects, for *Enterprise Resource Planning (ERP)* software. These application software projects consist of models, configuration data, and source code, and are in general organized as silos as well. Thus making it very difficult to collaborate between different projects. Using a metaprogramming environment that decouples models and coding templates in the way we have presented, could open up new possibilities.

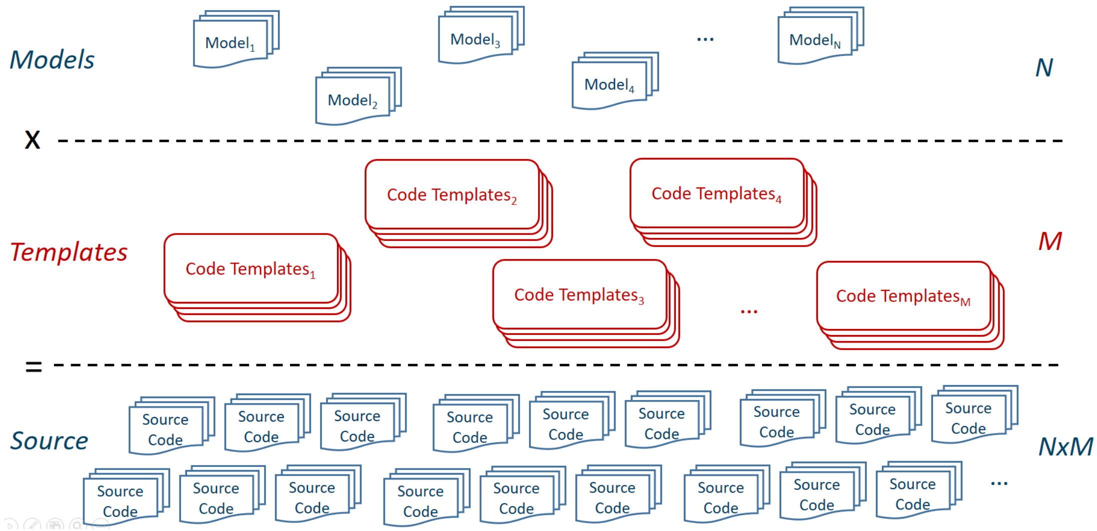


Fig. 9. A graphical representation of four variability dimensions within a Normalized Systems application codebase.

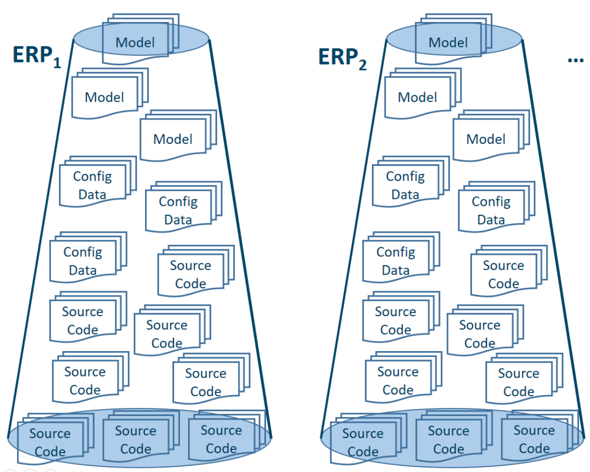


Fig. 10. Representation of various ERP software silos.

Modelers and designers would be able to collaborate on domain models, gradually improving existing model versions and variants, and adding on a regular basis new functional business modules, without having to bother about technicalities. (Meta)programmers would collaborate on coding templates, gradually improving and integrating new insights and coding techniques, adding and improving implementations of cross-cutting concerns, and providing support for modified and/or new technologies and frameworks. Application software systems would be generated to a large extent for a selected set of versions of domain models, using a specified set of coding templates, being targeted at a specific set of technology platforms. This would realize the above mentioned variation gains as described in Equations (2), (3), and (4).

C. Enabling Alternative Meta-Models

The current meta-model of our metaprogramming environment, consists of data, action, workflow, trigger, and connector elements, and is to a large extent specific for (web-based) information systems. It is not only conceivable to modify and/or extend this meta-model, but one could also imagine to define completely different meta-models for other purposes. Such meta-models could for instance be created to model traditional computing functions based on sequencing-selection-iteration, or to represent microservices or scripting units running parts of data mining algorithms in the cloud.

The meta-circular architecture presented in Figure 6 supports not only the definition of other models, but enables the definition of other meta-models as well. Indeed, the reader, model, control, and view classes could be generated for such an alternative meta-model, allowing the specification of models based on this new meta-model, both in XML or in a newly generated meta-application. These model parameters could then be propagated to new sets of coding templates. This could allow the presented meta-circular architecture to generate all types of source code or configuration data for all kinds of applications and languages.

VI. Some Results and Discussion

In this section, we present and discuss some empirical results regarding the use of the meta-circular expansion software. It is based on qualitative research that has been performed within the company, i.e., NSX bv, that develops the expansion software, and relies on nonnumerical data obtained from first-hand observation and interviews.

A. Refactoring of Existing Expanders

The Normalized Systems expander software has been in development since late 2011. Over the years, it was used by several organizations to generate, and re-generate on a regular basis, tens of information systems [18] [19]. During these years, and often on request of these organizations, many additional features and options were built into the source code templates. The overall refactoring was triggered by a concern over the growing size — and therefore complexity — of the model and control classes. It was also motivated by a desire to leverage the implicit homoiconicity of the NS (meta-)model to increase the productivity of improving and extending the expander software.

The complete refactoring was started in the last quarter of 2018, and performed in six months by two developers. Afterwards, the 182 expanders, each expanding or generating a specific source code artifact, were cleanly separated, and the software developers considered the expander codebase to be much better maintainable. Moreover, the learning curve for developers to take part in expander development was previously considered to be very steep, mainly due to the size and complexity of the model and control classes. After the refactoring, nearly all of the approximately 20 application developers of the company, have stated that the learning curve is considerably less steep, and that they feel comfortable to add features and options to the expander software themselves. One year later, most of them have indeed made such a contribution. The lead time of performing fixes and modifications to the expander coding templates, and to deliver it to the application project teams, has decreased from weeks to days.

B. Creating Additional Expander Bundles

Besides contributing to the original set of 182 expanders, application developers —even junior developers— are able to create additional *expander bundles* implementing a set of expanders for a specific functionality. Immediately after the refactoring, a junior developer has created in two months such a bundle of 20 expanders, targeted mainly at the implementation of REST services using Swagger. One year later, this bundle has been used successfully in numerous projects, and has grown significantly. Other —often junior— developers have created in the meantime various expander bundles providing valuable functionality. These include a bundle to generate mobile apps connecting to the expanded web-based information systems, a bundle to define more advanced search queries and reporting, and a bundle supporting various types of authentication. Recently, a newly hired graduate developed a small bundle of expanders to implement row-level security, only a couple of weeks after joining the company.

Currently, two different customers are developing expander bundles as well. As our goal is to establish a scalable collaboration in metaprogramming across a wide range of organizations and developers, we are setting up an exchange marketplace for expander bundles at *exchange.stars-end.net*.

C. Supporting Alternative Meta-Models

We have explained that the meta-circular metaprogramming architecture also provides the possibility to create and adopt other meta-models. Currently, a first implementation is available allowing to define other meta-models and providing coding templates, while all the code in between (readers, model, and control classes) of *this new* metaprogramming environment is automatically generated. A collaboration has been established with another group working on a metaprogramming environment, to perform a horizontal integration between the two metaprogramming environments. The first promising results of this integration have been reported [20].

One could argue that there is an implicit *meta-meta-model* underlying the various possible meta-models, and that this meta-meta-model could create a *silo-effect* hampering the integration of various metaprogramming efforts. However, the implicit meta-meta-model is based on the data elements of the NS meta-model, containing only data fields or attributes, and link fields or relationships. This is very similar, if not identical, to both the data entities of *Entity Relationship Diagrams (ERD)* with their data attributes and relationship links, and the entities or classes of the *Web Ontology Language (OWL)* with their datatype properties and object properties. Having for instance demonstrated the bi-directional transformation between our models and domain ontologies [21], we are confident that the dependency on the underlying meta-meta-model will not impede scalable collaborations.

VII. Conclusion

The increase of productivity and the improvement of evolvability are goals that have been pursued for a long time in computer programming. While more research has traditionally been performed on techniques to enhance productivity, our research on Normalized Systems Theory has been focusing on the evolvability of information systems. This paper presents a strategy to combine both lines of research.

While the technique of automated programming or source code generation was already part of our previous work on Normalized Systems, we have explored in this paper the incorporation of homoiconicity and meta-circularity to increase the productivity of our metaprogramming environment. A method was presented to turn the metaprogramming environment into a meta-circular architecture, using an homoiconic representation of the code generation models, and resulting in a considerable simplification of the expanders, i.e., the code generation software. We have argued that such a reduction of complexity could lead to a significant increase in productivity at the level of the development of the code generation software, and that the two-sided interfaces could enable a scalable collaboration on metaprogramming across organizations and developers. We have presented some preliminary results, indicating that the increase in metaprogramming productivity is indeed being realized, and have established a first collaborative integration effort with another metaprogramming environment.

This paper is believed to make some contributions. First, we show that it is possible to not only adopt code generation techniques to improve productivity, but to incorporate meta-circularity as well to improve both productivity and maintainability at the metaprogramming level. Moreover, this is demonstrated in a framework primarily targeted at evolvability. Second, we have presented a case-based strategy to make a code generation representation homoiconic, and the corresponding application architecture meta-circular. Finally, we have argued that the simplified structure of the code generation framework improves the possibilities for collaboration at the level of metaprogramming software.

Next to these contributions, it is clear that this paper is also subject to a number of limitations. It consists of a single case of making a code generation or metaprogramming environment meta-circular. Moreover, the presented results are both qualitative and preliminary, and the achieved collaboration on metaprogramming software is limited to a small amount of organizations. However, we are currently working to set up a collaboration of developers on a much wider scale at the level of metaprogramming, and to prove that this architecture can lead to new and much higher levels of productivity and collaboration in the field of automatic programming.

REFERENCES

[1] H. Mannaert, K. De Cock, and P. Uhnák, “On the realization of meta-circular code generation: The case of the normalized systems expanders,” in *Proceedings of the Fourteenth International Conference on Software Engineering Advances (ICSEA)*, November 2019, pp. 171–176.

[2] R. Agarwal and A. Tiwana, “Editorial—evolvable systems: Through the looking glass of IS,” *Information Systems Research*, vol. 26, no. 3, pp. 473–479, 2015.

[3] D. Parnas, “Software aspects of strategic defense systems,” *Communications of the ACM*, vol. 28, no. 12, pp. 1326–1335, 1985.

[4] P. Cointe, “Towards generative programming,” *Unconventional Programming Paradigms. Lecture Notes in Computer Science*, vol. 3566, pp. 86–100, 2005.

[5] J. R. Rymer and C. Richardson, “Low-code platforms deliver customer-facing apps fast, but will they scale up?” Forrester Research, Tech. Rep., 08 2015.

[6] B. Reselman, “Why the promise of low-code software platforms is deceiving,” TechTarget, Tech. Rep., 05 2019.

[7] A. Wortmann, “Towards component-based development of textual domain-specific languages,” in *Proceedings of the Fourteenth International Conference on Software Engineering Advances (ICSEA)*, 2019, pp. 68–73.

[8] K. Gusarovs and O. Nikiforova, “An intermediate model for the code generation from the two-hemisphere model,” in *Proceedings of the Fourteenth International Conference on Software Engineering Advances (ICSEA)*, 2019, pp. 74–82.

[9] U. Frank, “Specification and management of methods - a case for multi-level modelling,” *Business Process and Information Systems Modeling*, vol. 352, pp. 311–325, 2019.

[10] J. Hutchinson, M. Rouncefield, and J. Whittle, “Model-driven engineering practices in industry,” in *Proceedings of the International Conference on Software Engineering 2011*, 2011, pp. 633–640.

[11] J. Whittle, J. Hutchinson, and M. Rouncefield, “The state of practice in model-driven engineering,” *IEEE Software*, vol. 31, no. 3, pp. 79–85, 2014.

[12] H. Mannaert, J. Verelst, and P. De Bruyn, *Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design*. Koppa, 2016.

[13] C. Mooers and L. Deutsch, “Trac, a text-handling language,” in *ACM ’65 Proceedings of the 1965 20th National Conference*, 1965, pp. 229–246.

[14] D. McIlroy, “Macro instruction extensions of compiler languages,” *Communications of the ACM*, vol. 3, no. 4, pp. 214–220, 1960.

[15] J. Reynolds, “Definitional interpreters for higher-order programming languages,” *Higher-Order and Symbolic Computation*, vol. 11, no. 4, pp. 363–397, 1998.

[16] H. Mannaert, J. Verelst, and K. Ven, “The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability,” *Science of Computer Programming*, vol. 76, no. 12, pp. 1210–1222, 2011, special Issue on Software Evolution, Adaptability and Variability.

[17] ——, “Towards evolvable software architectures based on systems theoretic stability,” *Software: Practice and Experience*, vol. 42, no. 1, pp. 89–116, 2012.

[18] P. De Bruyn, H. Mannaert, and P. Huysmans, “On the variability dimensions of normalized systems applications: Experiences from an educational case study,” in *Proceedings of the Tenth International Conference on Pervasive Patterns and Applications (PATTERNS)*, 2018, pp. 45–50.

[19] ——, “On the variability dimensions of normalized systems applications: experiences from four case studies,” *International Journal on Advances in Systems and Measurements*, vol. 11, no. 3, pp. 306–314, 2018.

[20] H. Mannaert, C. McGroarty, K. De Cock, and S. Gallant, “Integrating two metaprogramming environments : an explorative case study,” in *Proceedings of the Fifteenth International Conference on Software Engineering Advances (ICSEA)*, October 2020, pp. 166–172.

[21] M. Suchánek, H. Mannaert, P. Uhnák, and R. Pergl, “Bi-directional transformation between normalized systems elements and domain ontologies in owl,” in *Proceedings of the fifteenth International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, May 2020, pp. 1–12.

The Matching Lego(R)-Like Bricks Problem: A Metaheuristic Approach

Martin Zinner*, Rui Song†, Kim Feldhoff*, André Gellrich†, Wolfgang E. Nagel*
* Center for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden
Dresden, Germany
E-mail: {martin.zinner1, kim.feldhoff, wolfgang.nagel}@tu-dresden.de
† Technical Information Systems
Technische Universität Dresden
Dresden, Germany
E-mail: {rui.song@tu-dresden.de, andre.gellrich@gmail.com}

***Abstract*—We formulate and transform a real-world combinatorial problem into a constraint satisfaction problem: choose a restricted set of containers from a warehouse, such that the elements contained in the containers satisfy some restrictions and compatibility criteria. We set up a formal, mathematical model, describe the combinatorial problem and define a (nonlinear) system of equations, which describes the equivalent constraint satisfaction problem. Next, we use the framework provided by the Apache Commons Mathematics Library in order to implement a solution based on genetic algorithms. We carry out performance tests and show that a general approach, having business logic solely in the definition of the fitness function, can deliver satisfactory results for a real-world use case in the manufacturing industry. To conclude, we use the possibilities offered by the jMetal framework to extend the use case to multi-objective optimization and and compare different heuristic algorithms predefined in jMetal applied to our use case.**

Keywords–Constraint satisfaction problem; Combinatorial problem; Genetic algorithm; Crossover; Mutation; Multi-objective optimization; Apache Commons Math.; jMetal.

I. INTRODUCTION

We formulate a new real-world combinatorial problem, the motivation for our study [1]. Initially, we describe succinctly the real-world problem as it has been identified at a semiconductor company and present the general strategy to solve it. In addition to the Single-Objective Optimization [1] using Genetic Algorithms based on the Apache Framework, we present a Multi-Objective Optimization strategy using Genetic Algorithms based on the jMetal Framework and compare the results. In order to avoid the technical difficulties related to the industrial application, we present the equivalent problem based on LEGO® bricks.

A. Motivation

Some time ago we were facing a strategic problem at a big semiconductor company. The company produces Integrated Circuits (ICs), also termed *chips*, assembles them to modules on a circuit board according to guidelines and specifications, and ships the modules as the final product to the customer. The ICs are stored in bins before the last technological process (cleaning) is performed.

The difficulties arise due to technical limitations of the tool that assembles the ICs to modules. The tool can handle at most five bins at once. This means in particular, that the ICs required to fulfill an order from the customer have to be in not more than five bins. Once the bins have been identified, the modules are assembled and shipped to the customer. If it is not possible to identify five bins in connection with a customer order, then either cost-intensive methods (rearranging the content of some bins) or time-intensive methods (waiting some days till the production process delivers new ICs) have to be applied. Hence, identifying the bins necessary to fulfill an order is crucial for the economic success of the company.

B. Current State and Challenge

There has been a selection algorithm in place, based primarily on heuristics and inside knowledge regarding the patterns of the specifications of the modules. Although the existing selection algorithm delivered satisfactory results in most of the cases, it runs for days in some cases and is not flexible enough, in particular, it cannot handle slight deviations from the existing specification patterns.

To circumvent the above inconvenient, the main aim of our study is to determine alternative selection methods, which always deliver satisfactory results within an acceptable time frame, and which are easy adaptable to meet future requirements. Our main objective is to identify and formalize the industrial problem as a mathematical model and to transform the occurring Combinatorial Problem (CP) into a Constrained Satisfaction Problem (CSP). The exact method using MATLAB did not deliver results within a satisfactory time frame. A suitable heuristic method – including Simulated Annealing (SA), Ant Colony Optimization (ACO), Genetic Algorithms (GA), etc. – to solve the CSP within the requirements had to be identified and appropriate algorithms had to be developed, which satisfy both the accuracy and performance demands.

If the general task is to find optimal solution to a set of constraints, we speak about Constrained Optimization Problem (COP). The primarily purpose of the industrial problem is to find a satisfactory solution, since from the technical perspective undercutting the requirements of the specifications does not lead to better quality. However, a straightforward extensions of the CSP towards COP is mentioned later.

C. Problem Description

The following example is artificial, it does not occur in *real life* in this manner, although it is very close to it. It is used to best describe the problem without burden the reader with the technical details of a concrete “real life” example. Later on, we will present a “real life” example from the industry and specify the respective mappings between the two models.

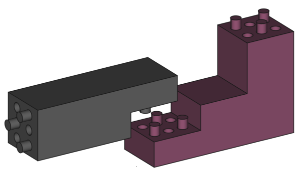


Figure 1: Illustration how two bricks, one of them a corner brick, can be pooled together.

We describe the problem succinctly by using an analogy of building structures out of LEGO®-like pieces (bricks). LEGO®-like pieces (also termed *blocks* or *bricks*) can be assembled to build sophisticated structures (in the following termed *objects*) like buildings, etc. Figure 1 shows how two bricks can be pooled together. The manufacturer of the bricks wants to facilitate and simplify the assembling of the bricks to the final objects as well as to cut manufacturing costs and establishes a two phases strategy when designing the layout plans of the final objects. The final object is parsed into components (termed *modules* or *assemblies*) in a straightforward way, such that these modules can also be reused to assemble other objects. This strategy of representing the final object as a composition of modules is very similar to the construction of buildings out of prefabricated structural components, i.e., modules. This way, by using a modular approach, the description and the design plans of quite sophisticated objects can be kept relatively simple and manageable and the complexity and the difficulty of building the final object is delegated to the assembly of the modules. Hence, the building specification of the final object is split into two guidelines, one regarding how to assemble the required modules, one regarding how to put together the modules to form the final object.

Each brick has numerical and non-numerical characteristics. A non-numerical attribute is, for example, a unique ID which characterizes the bricks like shape, approximate dimensions, number and the arrangement of the inner tubes, etc. Another non-numerical attribute is the color of the bricks, etc. There are very tight requirements in order to be able to assemble two or more bricks. In order to cut costs the technological process to manufacture the bricks is kept simple and cost-effective to the detriment of interchangeability. Thus, the pieces are measured after the production process and the measurement values are persisted in adequate storage systems.

In order to be able to assemble the bricks, they have to fit together, i.e., some measurement values (see Figure 2 for an example) have to fulfill some constraints. The respective measurement values must match in order that the bricks can be assembled. For example, putting four bricks together, side by side and on top of each other, strict restrictions concerning perpendicularity and planarity tolerance, have to be satisfied, such that for example, the overall maximum planarity error is 0.05 mm and the maximum perpendicularity error is 0.1 angular degree. Unfortunately, these restrictions can only be evaluated when all the measurement values of the bricks chosen to build the module are at the builder’s disposal. Corresponding calculation prescription are available.

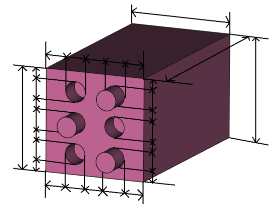


Figure 2: Exemplification of the measurements of a brick.

Once, the modules have been assembled, the object can be put together out of the pre-assembled modules with no limitations. Furthermore, all the modules are interchangeable with similar ones.

The manufacturing of the bricks is continuous, the bricks are packed into bins after the measuring process occurred and stowed in a warehouse. The ID, the non-numerical attributes and the numerical measurement values are stored in a database and associated to the bin ID. This way, the manufacturer knows exactly the content of each bin. In order to keep the manufacturing costs low, the bins are never repacked, after a bin is full and in the warehouse.

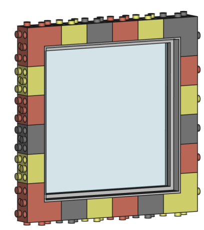


Figure 3: A frame with window as an example for a module.

The assembly plan for a particular structure (for example as in Figure 3) is not univocal, i.e., the number and the type of the bricks to build the envisaged structure is not unequivocally specified, the assembly plan contains more alternatives. Since the manufacturer provides detail information in digital form regarding each brick contained in the bins offered for sale, a computer program could easily verify that a house as given in Figure 4 could be built up from a particular set of bins. Unfortunately, identifying the set of bins necessary to build an object (for example the house as in Figure 4) turns out to be a very hard task to accomplish. In order to keep costs down, the number of the bins to be purchased, has to be limited to the necessary ones.

Let us suppose that the order can be assembled out of 5 bins, and the manufacturer offers 1000 bins for sale on his home page. Regrettably, the computer program can only verify if a particular set of five bins contains the bricks necessary to build the house. The brute force method to verify each set of 5 bins out of 1000 does not deliver a practical solution as elementary combinatorics show. Thus, other methods have to be applied.

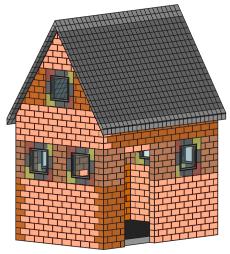


Figure 4: House as exemplification of an order composed of modules.

D. Outline

The remainder of the paper is structured as follows: Section II gives an overview about existing work related to the described problem. Section III introduces the mathematical model and describes how the combinatorial problem can be transformed into a constrained satisfaction problem. Section IV applies the proposed selection algorithm based on genetic algorithms to an industrial use case and shows the performance of an implemented solution which is based on genetic algorithms. A short investigation regarding multi objective optimization is considered in Section V, these investigations are extended by considering the porting and extension of the use case to the jMetal framework in Section VI. Additionally, the possibility offered by jMetal to compare different heuristic algorithms is taken advantage of, whereas Section VII concludes this paper and sketches the future work.

II. Related work

Generally speaking, combinatorial optimization problems are considered as being difficult [2] [3], which stimulated the development of effective approximate methods for their solutions. Combinatorial optimization problems appear in a multitude of real world applications, such as routing, assignment, scheduling, cutting and packing, network design, protein alignment, and in many fields of utmost economic, industrial, and scientific importance. The techniques for solving combinatorial optimization problems can be exact and heuristics. Exact algorithms guarantee optimal solutions, but the execution time often increases dramatically with the size of the underlying data, such that only small size of instances can be exactly solved. For all other cases, optimality is sacrificed for solvability in a limited amount of time [4].

The concept of a constraint satisfaction problem has also been formulated in the nineteen seventies by researchers in the artificial intelligence. Characteristic CSPs are the *n* queens problem, the zebra puzzle, the full adder circuit, the crossword puzzle, qualitative temporal reasoning, etc. Typical examples of constrained optimization problems are the knapsack problem and the coins problem [5]. Further examples of combinatorial optimization problems [6] are: bin packing, the traveling salesman problem, job scheduling, network routing, vehicle routing problem, multiprocessor scheduling, etc.

For the last decades, the development of theory and methods of computational intelligence regarding problems of combinatorial optimization was of interest of researchers. Nowadays, a class of evolutionary methods [7]–[10] is of particular interest, like simulated annealing, ant colony optimization, taboo search, particle swarm optimization, to which genetic algorithms belong [11]–[14]. Recent publications in this direction [15]–[21] prove the efficacy of applying genetic and other evolutionary algorithms in solving combinatorial optimization problems.

A genetic algorithm is an adaptive search technique based on the principles and mechanism of natural selection and of the survival of the fittest from the natural evolution. The genetic algorithms evolved from Holland's study [22] of adaptation in artificial and natural systems [6].

Typical examples of using evolutionary algorithms are the genetic algorithm approach to solve the hospital physician scheduling problem and an ant colony optimization based approach to solve the split delivery vehicle routing problem [23].

The report [24] offers an approach to use genetic algorithms to solve combinatorial optimization problems on a set of euclidean combinatorial configuration. The euclidean combinatorial configuration is a mapping of a finite abstract set into the euclidean space using the euclidean metric. The class of handled problems includes a problem of balancing masses of rotating parts, occurred in turbine construction, power plant engineering, etc.

III. The Formal Model

In the following, we will formalize the description of the combinatorial problem by introducing a mathematical model. This way, we use the advantages of the rigor of a formal approach over the inaccuracy and the incompleteness of natural languages. First, we introduce and tighten our notation, then we present the formal definition of the constraints which are considered in our formal model and which are the major components in the definition of the fitness function used to control and steer the genetic algorithm. Concluding, the combinatorial problem is defined as a constraint satisfaction problem.

A. Notation

In the following, we will formalize the description by introducing a mathematical model in order to use the advantages of the rigor of a formal approach over the inaccuracy and the incompleteness of natural languages.

Let $\mathfrak{V}$ be an arbitrary set. We notate by $\mathcal{P}(\mathfrak{V})$ the power set of $\mathfrak{V}$, i.e., the set of all subsets of $\mathfrak{V}$, including the empty set and $\mathfrak{V}$ itself. We notate by $\mathrm{card}(\mathfrak{V})$ the cardinality of $\mathfrak{V}$. We use a calligraphic font to denote index sets, such that the index set of $\mathfrak{V}$ is notated by $I^{\mathcal{V}}$.

The finite sets of bricks, bins, (non-numerical type of) attributes, (numerical type of) and measurements are denoted as follows:

$$\begin{aligned}\mathfrak{S} &:= \{s_i \mid i \in I^{\mathcal{S}} \text{ and } s_i \text{ is a brick (stone)}\},\\ \mathfrak{B} &:= \{b_i \mid i \in I^{\mathcal{B}} \text{ and } b_i \text{ is a bin (carton)}\},\\ \mathfrak{A} &:= \{A^i \mid i \in I^{\mathcal{A}} \text{ and } A^i \text{ is an attribute}\},\\ \mathfrak{M} &:= \{M^i \mid i \in I^{\mathcal{M}} \text{ and } M^i \text{ is a measurement}\}.\end{aligned}$$

Let $i \in I^{\mathcal{A}}$, $j \in I^{\mathcal{M}}$, and $k \in I^{\mathcal{S}}$. We denote by a_k^i the value of the attribute A^i of the brick s_k and by m_k^j the value of the measurement M^j at the brick s_k. We denote the list of assembly units (modules) by

$$\mathfrak{U} := \{U^i \mid i \in I^{\mathcal{U}} \text{ and } U^i \text{ is an assembly unit (module)}\}.$$

The construction (guideline) plan for a module $U \in \mathfrak{U}$ contains

a) the (three dimensional) design plan description, i.e., the position of each brick within the module,
b) the non-numerical attribute values for each brick and
c) prescriptions regarding the measurement values.

Analogously, we denote by

$$\mathfrak{O} := \{O^i \mid i \in I^O \text{ and } O^i \text{ is an object}\}$$

the list of objects for which there exists construction plans.

The non-numerical attributes values of the selected bricks have to match the corresponding values in the guideline plan.

We denote by

$$\mathfrak{R} := \{R^i \mid i \in I^{\mathcal{R}} \text{ and } R^i \text{ is a requirement (specification)}\}$$

the list of the requirements (specifications) of the objects.

B. Transformation of the CP into a CSP

Let $O \in \mathfrak{O}$ an order. Then, according to the specifications, there exists *(proxy) modules* $\hat{M}^{l_1}, \hat{M}^{l_2}, \ldots, \hat{M}^{l_k}$, such that O is an ordered list of modules, i.e., $O = (\hat{M}^{l_1}, \hat{M}^{l_2}, \ldots, \hat{M}^{l_k})$. The term *proxy* is used to denote an abstract entity according to the specifications. Analogously, each module $\hat{M}^i$ with $i \in \{l_1, l_2, \ldots, l_k\}$ is an ordered list of *proxy bricks* (stones), i.e., $\hat{M}^i = (\hat{s}_{i_1}, \hat{s}_{i_2}, \ldots, \hat{s}_{i_m})$. Hence, each module can be represented by an ordered list of proxy bricks, i.e., $O = (\hat{s}_{k_1}, \hat{s}_{k_2}, \ldots, \hat{s}_{k_n})$. This representation will be used later to define the individuals within the context of the genetic algorithms. Furthermore, we set as $\mathrm{card}(O)$ the number of proxy bricks associated to the order O, i.e., $\mathrm{card}(O) = n$.

Let $O = (\hat{s}_{k_1}, \hat{s}_{k_2}, \ldots, \hat{s}_{k_n})$ be an order. We say that the ordered list $(s_{k_1}, s_{k_2}, \ldots, s_{k_n})$ with $s_i \in \mathfrak{S}\ \forall i \in \{k_1, k_2, \ldots, k_n\}$ is an assignment (embodiment) of O. This means especially that the abstract unit of the specification is materialized within the production process. We set

$$\mathfrak{E} := \{E^i \mid i \in I^{\mathcal{E}} \text{ and } E^i \text{ is an assignment (embodiment)}\}.$$

Let $R \in \mathfrak{R}$ be a requirement (specification) of a specific module $U \in \mathfrak{U}$ and let $\{\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_n\}$ be the proxy bricks of the specification. The design plan of the specification provides the three-dimensional assembly plan of the proxy bricks. Additionally, the specifications provide information regarding the restriction the bricks have to fulfill in order to be eligible. For each $j \in \{1, 2, \ldots, n\}$ the specifications contain the values $\{\hat{a}^i_j \mid i \in I^{\mathcal{A}}\}$ of the attributes $\mathfrak{A} := \{A^i \mid i \in I^{\mathcal{A}}\}$ at the proxy brick $\hat{s}_j$. We use the symbol $\hat{a}^i_j$ to denote the value of the attribute a^i of the proxy (placeholder) brick $\hat{s}_j$.

This means especially, that the brick s_j can substitute the proxy brick $\hat{s}_j$ if the values of the corresponding attributes coincide, i.e., $a^i_j = \hat{a}^i_j$ for all $i \in I^{\mathcal{A}}$.

More formally, the attributes must satisfy certain constraints:

$$C_A : \mathfrak{A} \times \mathfrak{S} \to \{yes, no\},$$
$$\{s^i_j \mid i \in I^{\mathcal{A}},\ j \in I^{S}\} \mapsto C_A(a^i_j).$$

$C_A(a^i_j) = yes$ if the attribute constraint is satisfied for the brick s_j i.e., $a^i_j = \hat{a}^i_j$, else $C_A(a^i_j) = no$.

On the other side, the (numerical) measurement values must also satisfy certain constraints (restrictions). For example, the standard deviation of the respective measurement values for some bricks of a specific module should not surpass some given limits. Formally, C_M can be represented as:

$$C_M : \mathfrak{M} \times \mathfrak{U} \to \{yes, no\},$$
$$\{m^i_j \mid i \in I^{\mathcal{M}},\ j \in I^{\mathcal{U}}\} \mapsto C_M(m^i_j).$$

$C_M(m^i_j) = yes$ if the constraint is satisfied for the bricks belonging to the module U^j, else $C_M(m^i_j) = no$.

In order to be able to reduce the constraints to brick level, i.e., to be able to decide whether the constraint is satisfied for a specific brick or not, we use the restriction C^S_M of C_M namely $C^S_M := C_M\big|_S$ such that $C^S_M(s^i_j) = yes$ if $s_j \in U^j$ and $C_M(m^i_j) = yes$; else $C^S_M(m^i_j) = no$. Let $U \in \mathfrak{U}$ be a module. The above means especially, that the measurement constraint on brick level is satisfied for $s \in U$ if the measurement constraint is satisfied (on module level) for U.

Since the measurement values do not really characterize the modules (they must only fulfill the requirements regarding the constraints), we introduce equivalence classes on the set of modules. Two modules belong to the same class if

a) they have both the same design plan,
b) the component bricks fulfill the same (non-numerical) attributes and
c) the prescriptions regarding the measurement values are satisfied for both modules.

Accordingly, two modules belonging to the same equivalence class are interchangeable.

Hence, all the bricks needed for a module must be selected in order to be able to finally decide if the constraints are satisfied or not.

As already mentioned, each object $O \in \mathfrak{O}$ should be assembled out of bricks contained in a reduced number of bins. We set $MaxB^O$ for the maximum number of bins as mentioned above.

Let $i \in I^{\mathcal{B}}$, $j \in I^O$, let $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ be the content of the bin b_i and let $\{s_{j_1}, s_{j_2}, \ldots, s_{j_l}\}$ be an assignment of $O^j \in \mathfrak{O}$. We set $\overline{b}^j_i := 1$ if $\{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\} \cap \{s_{j_1}, s_{j_2}, \ldots, s_{j_l}\} \neq \emptyset$ else 0. This means especially, that $\overline{b}^j_i := 1$ if the bin contains bricks belonging to the respective assignment of O^j.

Analogously, the constraints regarding the bins (cartons) can be regarded formally as:

$$C_B : \mathcal{P}(\mathfrak{B}) \times \mathfrak{O} \to \{yes, no\},$$
$$\{\mathbb{B} \in \mathcal{P}(\mathfrak{B}),\ O^j \in \mathfrak{O}\} \mapsto C_B(\mathbb{B}, O^j).$$

Let $\mathbb{I}$ an index set, such that $\mathbb{B} = \{b_i | i \in \mathbb{I}\}$. Then $C_B(\mathbb{B}, O^j) = yes$ if

$$\sum_{i \in \mathbb{I}} \bar{b}_i^j \leq MaxB^{O^j}, \tag{1}$$

i.e., the bricks of the order O^j are contained in no more than $MaxB^{O^j}$ bins. Additionally, $C_B(\mathbb{B}, O^j) = no$ if the above condition is not satisfied.

Similar to the measurement constraint C_M, we reduce C_B to brick level. Let $\mathbb{S} := \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$ be an assignment of O^i. Let $\mathbb{B} = \{b_{l_1}, b_{l_2}, \ldots, b_{l_n}\}$ be a set of bins, such that each bin contains at least one $s \in \mathbb{S}$ and there is no brick $s \in \mathbb{S}$ which is not contained in one of the bins of the set $\mathbb{B}$. In this sense, $\mathbb{B}$ is minimal regarding the assignment $\mathbb{S}$. Then, for the restriction C_B^S on S of C_M we have $C_B^S(s) = yes$ if $C_B(\mathbb{B}, O^i) = yes$, i.e., all the bricks of the assignment $\mathbb{S}$ are stored in no more than $MaxB^{O^j}$ bins. Additionally, $C_B^{S,U}(s) = no$ if $C_B(\mathbb{B}, O^i)$ is not satisfied.

Until now, we considered the constraints related to the architecture of the object, i.e., related to the attributes of a particular brick, the measurement values of the bricks belonging to a module, and the restrictions regarding the bins which contain the bricks. We can condense the constraints mentioned above, such that they relate only to bricks. This means especially, that the measurement constraint are satisfied for a brick, if there is a group of bricks (module, or order), such that the given measurement constraint is satisfied as described above.

We set accordingly:

$$\mathfrak{C} := \{C^i \mid i \in I^C \text{ and } C^i \text{ is a distinct constraint}\}$$

the list of distinct constraints.

The constraints can be considered as a function. Please recall that I^S is the index set of $\mathfrak{S}$.

$$C : \mathfrak{S} \times I^C \to \{yes, no\},$$
$$\{s_k^i \mid i \in I^C,\, k \in I^S\} \mapsto C(s_k^i).$$

Please consider, that the above representation can be misinterpreted, such that the constraint is exclusively a property of the respected brick. This is not the case, for example the measurement constraints fulfilled or not for the bricks assigned to a module. Hence, if one brick is changed, then the constraints of all the bricks belonging to a module can be invalidated.

We define now formally the weights, (i.e., w is a weight if $w > 0$) which are necessary to be able to model the importance of the constraints within the genetic algorithm. We set

$$\mathfrak{W} := \{w_i \mid i \in I^C \text{ and } w_i \text{ is a weight}\}$$

the list of weights. This means especially, that each constraint has an associated weight.

The fitness function [25] characterizes the quality of an assignment of an order, such that a value closer to 1 means a better quality. It plays an important role in the decision, whether an assignment fulfills the specifications or not.

The purpose of the following function inv is purely technical, it is used to switch the values of the boolean values 1 and 0 to be used in the definition of an example of the fitness function, i.e., $inv : \{yes, no\} \to \{0, 1\}$ such that $inv(yes) = 0$ and $inv(no) = 1$.

Please find below an example for the fitness function. Let $I^{\mathbb{S}} \subset I^S$ and let $w_i \in \mathfrak{W}$ for all $i \in I^C$. Then:

$$F : \mathfrak{E} \times I^C \to (0, 1],$$
$$\{s_k^i \mid i \in I^C,\, k \in I^{\mathbb{S}}\} \mapsto \frac{1}{1 + \sum\limits_{i \in I^C, k \in I^{\mathbb{S}}} w_i \cdot inv(C(s_k^i))}. \tag{2}$$

Problem formulation (Combinatorial problem)

Let $O \in \mathfrak{O}$ a given object and let $n \in \mathbb{N}$.

Choose n bins from the warehouse, such that the object can be assembled out of the bricks contained in these bins according to the existing construction plans.

The construction plan for an object $O \in \mathfrak{O}$ specifies the lists of (non equivocally determined) modules, including the design plan, such that the object can be build out of these modules. Hence, it can be unambiguously decided, whether the n cartons contain the necessary bricks to assemble them to modules, which can be put together to form the required object. Let us suppose that $n \ll \mathrm{card}(\mathfrak{C})$, i.e., the number of bins in the warehouse exceeds the number of bins to be chosen by orders of magnitude. The difficulties of solving the problem in a straightforward way lie in the very large number of possibilities to combine n bins out of $\mathrm{card}(\mathfrak{B})$. Therefore, other strategies have to be used.

To summarize: the specification of an object (for example a house composed of bricks), contains very strict requirements regarding the components. The assembly plan specifies the strict order in which the bricks have to be assembled. Hence, the bricks must satisfy some attributes (like shape, type, color, etc., in order to satisfy the requirements of the construction plans. Moreover, some bricks have to fit together (for example the window frame) so they can be assembled in the order given by the construction plans. If the above requirements are satisfied for all the units (modules), the object can be assembled. Furthermore, the selected bricks have to be selected from a restricted number of bins (cartons). The latter makes the task so difficult.

From a formal point of view, the associated constraint satisfaction problem of the combinatorial problem can now be formulated:

Problem formulation (Constraint satisfaction problem)

Let $O \in \mathfrak{O}$ be an order with the representation $O = (\hat{s_1}, \hat{s_2}, \ldots, \hat{s_k})$. Set $w_i = 1$ for all $i \in I^C$.

Find an index set $\{l_1, l_2, \ldots, l_k\} \subset I^S$ such that $(s_{l_1}, s_{l_2}, \ldots, s_{l_k})$ is an assignment of O, having $F((s_{l_1}, s_{l_2}, \ldots, s_{l_k})) = 1$.

C. Formulation as a System of Equations

In the following, we will formulate the problem as a System of Equations. For readability reasons, we will refresh the some of the notations already introduced. Let $\mathbb{S} \subset \mathfrak{S}$ a subset of the bricks. We denote – as we have done before – by $I^{\mathbb{S}}$ the index set of $\mathbb{S}$. We recall that $I^{\mathcal{S}}$ is the index set of $\mathfrak{S}$, i.e., $I^{\mathbb{S}} \subset I^{\mathcal{S}}$ and I^C is the index set of the distinct constraints $\mathfrak{C}$. Moreover, we recall that C_B are the bin constraints and correspondingly, I^{C_B} is the index set of the bin constraints.

Let $O^j = (\hat{s}_{k_1}, \hat{s}_{k_2}, \ldots, \hat{s}_{k_n})$, $j \in I^O$ be an order. Then $\mathrm{card}(O^j)$ is equal to the number of proxy bricks, i.e., $\mathrm{card}(O^j) = n$. Let $i \in I^{\mathcal{B}}$. Please recall that $\overline{b}_i^j := 1$ if the bin b_i contains bricks belonging to the respective assignment of O^j.

Problem reformulation (Constraint satisfaction problem)
Find an assignment $\mathbb{S} \subset \mathfrak{S}$ of O^j such that all constraints are satisfied:

$$\sum_{i \in I^C,\ k \in I^{\mathbb{S}}} inv(C(s_k^i)) = 0. \tag{3}$$

Requiring that the sum of chosen bins is minimal, one can formulate the problem as an Optimization Problem (OP)

Problem formulation (Optimization problem)
Find an assignment $\mathbb{S} \subset \mathfrak{S}$ of O^j such that:

1) *all but the bin constraints are satisfied:*

$$\sum_{i \in (I^C \setminus I^{C_B}),\ k \in I^{\mathbb{S}}} inv(C(s_k^i)) = 0 \tag{4}$$

2) *and the number of bins is minimal:*

$$minimize \sum_{i \in \mathbb{I}} \overline{b}_i^j \tag{5}$$

Please recall that according to the description of the assignment, the equality

$$\mathrm{card}(\mathbb{S}) = \mathrm{card}(O^j) \tag{6}$$

holds, i.e., the number of the bricks we seek is determined through the specification of the order O^j.

Equation (3) means especially that all constraints – including the bin constraint (i.e., relation (1)) – have to be satisfied. In contrast to the above, the optimization strategy does not require that the bin constraint is satisfied (Equation (4)), but the minimal number of bins (see condition (5)) is seeked.

There are either no solution to CSP / OP or one / multiple solutions. The multiple solutions are equivalent, i.e., two solutions which satisfy the constraint are regarded as of the same quality. Accordingly, only one assignment is seeked.

Let A be an integer matrix; b and c vectors, an Integer Linear Program (ILP) is expressed as [26]

$$\max_{x \in \mathbb{Z}^n} \{c^T x \mid Ax \leq b,\ x \geq 0\}. \tag{7}$$

Unfortunately, the measurement constraints do not apply on the brick level, moreover it has to be decided for a module (i.e., a bunch of bricks) if the measurements values are satisfied. This means especially, that the measurement constraints are satisfied for all of the bricks of the module or for none of them. This means that the reformulated CSP cannot be represented in a direct way as an ILP.

However, in order to linearize the relation (4) the OP can be formally transformed by considering modules instead of bricks. Accordingly, a module M satisfies a specific constraint if all the bricks composing the module M satisfy the constraint. Hence, the module is contained in a nonempty set of bins and not anymore in a single bin. Unfortunately, this strategy can be applied only on very small data set, since the measurement constraints must verified for all modules which can be build out of the existing bricks.

To summarize, it does not seem that the relation (4) can be meaningfully linearized – due to the constraints which act on a bunch of bricks – in such a way that the reformulated CSP can be transformed into a ILP, thus benefiting from the advantages of the research in this area.

D. Reformulation as a Nonlinear System of Equations

In the following, we will reformulate the problem as a nonlinear system of equations. Therefore, firstly we recall and strengthen the relating definitions and correlations between them in order to ease the reading of the following section.

- $I^{\mathcal{B}}$ is the index set of the set of bins (cartons); obviously $\mathrm{card}(I^{\mathcal{B}}) = \mathrm{card}(\mathfrak{B})$, where $\mathfrak{B}$ is the set of bins (cartons),
- $I^{(\mathcal{S}^i)}$ is the index set of the bricks (stones) contained in bin b_i for all $i \in I^{\mathcal{B}}$,
- $\mathfrak{S}^i := \{s_{i,k} \mid i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}$ and $s_{k,i}$ is a brick (stone) contained in bin $b_i\} \subseteq \mathfrak{S}$, i.e., $\mathfrak{S}^i$ is the set of bricks contained in bin b_i, for all $i \in I^{\mathcal{B}}$, It follows from the definition that $\mathrm{card}(I^{(\mathcal{S}^i)}) = \mathrm{card}(\mathfrak{S}^i)$
- Clearly, $\mathrm{card}(\mathfrak{S}) = \sum_{i \in I^{\mathcal{B}}} \mathrm{card}(I^{(\mathcal{S}^i)})$, the total number of bricks (stones) is obtained by summing up the bricks contained in the cartons.

As mentioned, we denote by $s_{i,k}$ the brick located at bin b_i having the local index k and introduce the following set of boolean variables:

$$\overline{s}_{i,k} := \begin{cases} 1 & \text{if brick } s_{i,k} \text{ has been selected,} \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for all } i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}.$$

Please recall that $\mathfrak{U}$ is the set of all assembly units (modules), i.e., $\mathfrak{U} := \{U^i \mid i \in I^{\mathcal{U}}\}$; such that $I^{\mathcal{U}}$ is the index set of $\mathfrak{U}$. We denote:

$$\overline{s}_{i,k}^{U^j} := \begin{cases} 1 & \text{if } \overline{s}_{i,k} = 1 \text{ and brick } s_{i,k} \in U^j, \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for all } i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}, j \in I^{\mathcal{U}}.$$

and

$$\overline{s}_i := \begin{cases} 1 & \text{if } \exists k \in I^{(\mathcal{S}^i)} \text{ such that } \overline{s}_{i,k} = 1, \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for all } i \in I^{\mathcal{B}}.$$

As already defined, $I^{\mathcal{A}}$ be the index set of the set of the non-numerical attributes $\mathfrak{A}$. The matching of the non-numerical attributes is modeled by introducing a new set of boolean variables:

$$\overline{d}_{i,k}^{g} := \begin{cases} 1 & \text{if } f_{i,k}^{g} = \hat{f}_{i,k}^{g}, \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for all } g \in I^{\mathcal{A}}, i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}.$$

Unfortunately, the measurement constraints are assembly unit (module) related, i.e., the measurement values of the bricks belonging to a specific assembly unit as a whole satisfy or not the constraints requirements. For example, the standard deviation of a specific measurement may not surpass a given threshold value for the bricks belonging considered assembly unit. Accordingly, all bricks belonging to a module have to be selected before measurement satisfaction decisions can be validated.

Let $U \in \mathfrak{U}$ a particular assembly unit (module). The fulfilling of the constraints is modeled by introducing a new set of boolean variables:

$$\overline{e}_{i,k}^{h,U} := \begin{cases} 1 & \text{if } \forall s_{i,k} \in U : s_{i,k} \text{ fulfills constraint } E_h, \\ 0 & \text{otherwise} \end{cases}$$
$$\text{for all } h \in I^{\mathcal{M}}, i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}.$$

This means in particular, that opposite to the non-numerical attributes, the measurement constraints do depend on the specifications of each module.

Let $O \in \mathfrak{O}$ an order. Then, according to the definition of the order, there exists (proxy) modules $\{\hat{U}^i \mid i \in I^{\mathcal{U}_O}\}$, such that O is a ordered list of those modules. Let $I^{\mathcal{U}_O}$ the index set of the set of modules and let $MaxB^O$ the maximal number of bins where the bricks may be located. For the sake of simplifying the notations, we set $p := card(O)$ as the number of the bricks we seek to determine. By defining a vector $t \in \{0,1\}^{\text{card}(I^{\mathcal{B}}) \times \text{card}(I^{(\mathcal{S}^i)})}$ with elements $t_{i,k} := \overline{s}_{i,k}$ $\forall i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}$, the problem can be reformulated as a system of equations:

Problem reformulation (Non-linear system of equations)

Given all quantities listed above except for $\overline{s}_{i,k}$, $i \in I^{\mathcal{B}}, k \in I^{(\mathcal{S}^i)}$, *find* $t \in \{0,1\}^{\text{card}(I^{\mathcal{B}}) \times \text{card}(I^{(\mathcal{S}^i)})}$ *such that*

1) *p bricks from all cartons are chosen:*

$$\sum_{i \in I^{\mathcal{B}}} \sum_{k \in I^{(\mathcal{S}^i)}} \overline{s}_{i,k} = p, \tag{8}$$

2) *maximal $MaxB^O$ bins are chosen:*

$$\sum_{i \in I^{\mathcal{B}}} \overline{s}_i \leq MaxB^O, \tag{9}$$

3) *the non-numerical features are matched:*

$$\sum_{i \in I^{\mathcal{B}}} \sum_{k \in I^{(\mathcal{S}^i)}} d_{i,k}^{g} \cdot \overline{s}_{i,k} \geq p \quad \textit{for all } g \in I^{\mathcal{A}}, \tag{10}$$

4) *the measurement constraints are fulfilled:*

$$\sum_{i \in I^{\mathcal{B}}} \sum_{k \in I^{(\mathcal{S}^i)}} e_{i,k}^{h} \cdot \overline{s}_{i,k}^{U^j} \geq p \quad \textit{for all } h \in I^{\mathcal{M}}, j \in I^{\mathcal{U}}, \tag{11}$$

The inequations (10) result as follows: The attributes have to match for at least p bricks, thus, the product $d_{i,k}^{g} \cdot \overline{s}_{i,k}$ of at least p terms has to be one. Each of the remaining terms will be either one or zero due to the boolean variables. Therefore, the total sum will be also greater than or equal to p. The inequations (11) have been determined analogously. In total, there are one equation from (8), one inequation from (9), $\text{card}(I^{\mathcal{A}})$ inequations from (10), and $\text{card}(I^{\mathcal{M}}) \cdot \text{card}(I^{\mathcal{U}})$ inequations from (11) compared to $\sum_{i \in I^{\mathcal{B}}} card(I^{(\mathcal{S}^i)}) := card(I^{\mathcal{S}})$ unknown variables.

All equations are linear. Please note that the variables $s_{i,k}$ are boolean ones and thus, have to be integer. We only have to find one solution of the system. However, if we want to maximize the number of objects which can be assembled from the given set of cartons, then we can formulate an optimization problem based on the linear system defined above. It has to be examined if this optimization problem leads to an integer linear program (ILP). In case this is possible, then rewriting the ILP in standard form will be the basis for solving the problem.

There exists a lot of methods for solving ILPs. Branch-and-bound and branch-and-cut algorithms are two of them. They work in a heuristic way.

IV. USE CASE: AN EXCERPT

In the following, we present a real-life use case [27] we came across at an international semiconductor company. We describe the problem by using the specific terminology in the semiconductor industry, utilizing them with care and only when it is inevitable and undeniable necessary. We describe the fundamentals of the genetic algorithms and show the way it is used to solve our problem. Finally, we conclude by presenting some performance tests.

A. Problem Description

The company manufactures integrated circuits (ICs, also termed chips), which are subsequently assembled on circuits boards to salable entities, termed modules. In order to keep production cost low, the specification of the ICs do not impose very tight constraints on the attributes of the ICs, such that the same IC can be used for different types of modules. On the contrary, the specification regarding the modules are very stringent, in order that the module should be fully functional at the customer side. As soon as the ICs are manufactured, a good dozen of electrical properties are measured and persisted in a data repository. Figure 5 shows a symbolic representation of an IC.

Usually, four to six ICs are assembled on the module. The specification of the modules contains the design (i.e., number and positioning) of the ICs on the integrated circuit board, the type of the IC (article, number of pins, etc.), and several constrains regarding the interaction of the ICs of the module. Figure 6 shows a symbolic representation of a module with

5 ICs. In order for the module to be fully functional, the corresponding measurement values of the ICs have to be in a narrow range. For example, for a specific measurement, the values of the voltage of the ICs have to be between 2.1 volt and 2.5 volt in order that the IC is not scrapped and can be used for further processing. Unfortunately, not all the ICs having the corresponding measurement value in the range as described above, can be assembled to a module. The values differ too much from each other, and the module will not work properly at the customer side. To circumvent this impediment further constraints are needed. These constraints apply on all ICs of the module or just on a subset of it. For example, an often used constraint is limiting the standard deviation of the voltage to 0.1 within one module.

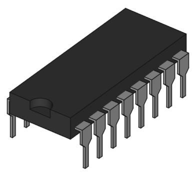


Figure 5: Symbolic representation of an IC.

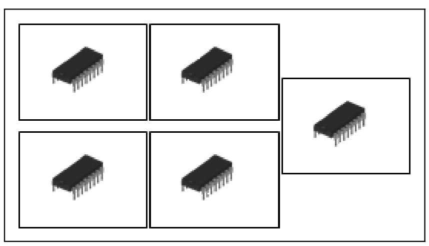


Figure 6: Symbolic representation of a module with 5 ICs.

The ordering unit (termed *work order*) contains the description of the modules, the customer expects to be shipped together at once. There are no additional constraints on the ICs regarding the work order. As soon as the manufacturing process of the ICs has been finished, the ICs are packed in boxes. That way the cleaning of the ICs can be performed and the ICs can be assembled to modules. The boxes are then transferred to the warehouse. Figure 7 shows a symbolic representation of the bins in the warehouse.

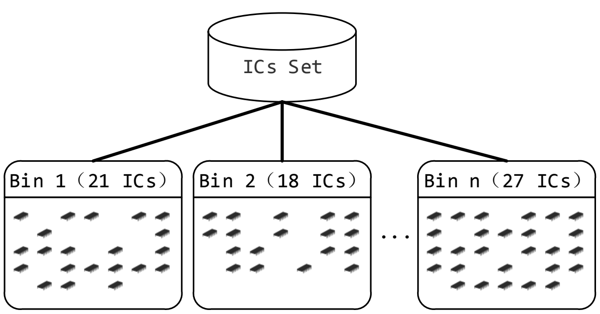


Figure 7: Symbolic representation of the Warehouse containing the bins and the bins containing the ICs.

The difficulties of the semiconductor company to honor each the order in general are also due to technological restrictions, since the tool that assembles the ICs can handle at most five boxes, i.e., all the ICs necessary to fulfill a work order should be located in five boxes. Due to the fact that the work orders always contain the same number of ICs, independent of the specification of the modules, the minimal number of boxes which are needed to meet the requirement of the work order is four with 9 percent surplus of ICs.

If we rephrase the above in a more concise form, the challenge is: Find five boxes in the warehouse, such that it contains the ICs needed to fulfill the requirements for a work order.

B. Used Methods

Simple combinatorics show that the brute force method, i.e., go through all the possibilities and check if the selected boxes fulfill the requirements, is not implementable for practical systems. Fortunately, there is an implementation in place for the selection strategy, based on heuristics, local optimum, and inside knowledge of the pattern of the modules. This way, we have a very good way to compare the results of the genetic algorithm with alternative solutions. Regretfully, our attempt to deliver exact solutions on the problem using MATLAB were not crowned by success due to the large amount of data and to the restricted computing power of the machines we used. The disadvantages of the already existing solution for the selection strategy were partly also the issues that made it possible to set up such a solution:

a) the unpredictability that the selection strategy delivers a solution within the expected time frame;

b) the inflexibility to even minor changes in the design and specifications of the modules, thus, the unpredictability that the software can be used in the future;

c) heavy maintenance efforts due to the sophisticated and architecture and implementation;

d) lack of the proprietary knowledge and documentation of the implementation on the low level side;

e) impossibility to reuse the existing code with reasonable efforts for further development and enhancements.

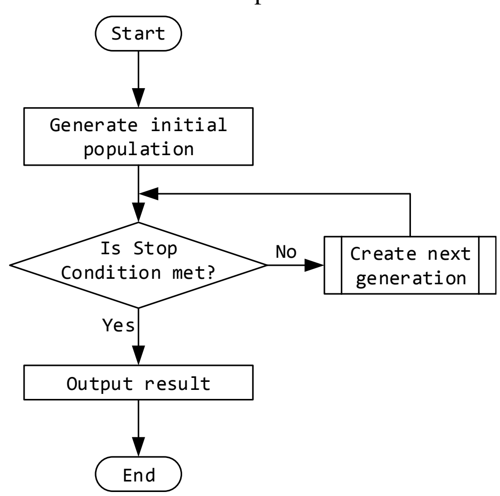


Figure 8: Overview of the genetic algorithm.

The concepts of the genetic methods are straightforward and easy to understand. The main idea is that we start with an initial population of individuals, and as time goes by, the genes of the descendants are improved, such that at least one individual satisfies the expectations. The individual incorporates the requirements of the problem. The expectation in the end is that these requirements are finally satisfied. Each individual owns genes, part of it is inherited by his descendants.

The principle of the genetic algorithm is straightforward, see Figure 8 for a basic representation. First, the initial population is randomly generated, then subsequent generations are created as long as the stopping condition is satisfied. The individuals with the highest quality characteristics constitute the main part of the output.

We define the individuals as an abstraction of the work order, such that each gene of the individual is the abstraction for an IC of the warehouse. Accordingly, the individual satisfies the requirements if the ICs can be assembled to modules, such that the corresponding work order is fulfilled.

Figure 9 is a symbolic representation of an individual (also termed chromosome) containing three modules. According to the specifications, each module contains six ICs. The slots of the modules – i.e., the plug-in location where the ICs are inserted – are numbered consecutively, from 1 to 18 (upper row of the table). Generally speaking, the specific slots – each of them corresponds to the specification of the components of the respective module – represent formally the requirements regarding the ICs composing the modules, hence the structure of the consecutive modules (Module 1 till Module 3) also determines the layout of the individual. Accordingly, the ICs assigned to the slots are the genes of the individual. The middle row of the table in Figure 9 shows an assignment of the slots with ICs, for example, the IC with the unique identifier (Id) 263 has been assigned to slot No. 1, etc. The bottom row illustrates the storage of the ICs in the bins, for example the IC with the Id 263 is stored in the bin No. 6. This latter allocation cannot be changed during the selection process, i.e., it is not possible to rearrange the content of the bins before or during the selection process.

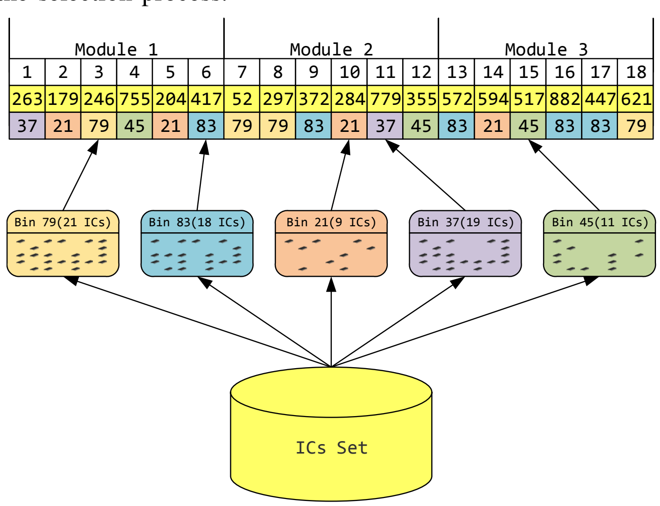


Figure 9: Symbolic representation of the module based structure of an individual (chromosome) containing the respective ICs (genes).

The initial population is randomly generated out of the ICs in the warehouse. The criterion, which determines to what degree the individual fulfills the requirement of the associated work order, is the fitness function. The fitness function takes values between 0 and 1, a greater value means that the individual is more close to fulfill the specification of the work order. To achieve a value of 1 is the ultimate goal. It means that the corresponding individual satisfies the requirements to fulfill the associated work order. Hence, the definition of the fitness function is one of the most sensible parts of the genetic algorithms and the setup of this function should be considered very carefully.

Actually, the strategy of the genetic algorithm resembles very much to the evolution of the mankind. People marry, have children by passing their genes to them, divorce and remarry again, have children, and so on and so forth. The expectation is that the descendants have more “advantageous genes”, regardless of how the term “advantageous genes” is defined.

Establishing the fitness function is one of the most important strategical decision to be taken when setting up the genetic algorithm. In our case, there are a few constraints (more than one) which affect the quality of the individuals. Implementations which try to find a Pareto optimal state [28], [29] (i.e., a state from which it is impossible to make an individual better, without making at least one individual worse) use strategies as tournament selection [30] or the improved Pareto domination tournament [29].

As already mentioned, the starting population is selected aleatorically. Once, the first generation is constituted, the preparations to generate the next generation are met. Unfortunately, the Apache Commons Mathematics Library does not support multi-objective optimization problems, hence our algorithms cannot use the strategy of the *Niched Pareto Genetic Algorithm* (NPGA) [29].

Instead, for each individual, the fitness function is calculated such that the suitability to fulfill the expectations, is evaluated for each individual. The higher the computed value is, the better fitted are the individuals. Let us suppose, that the initial population is composed of 500 individuals. We use some of the concepts provided by Apache Commons Mathematics Library in our implementation, among others the *elitism rate,* which specifies the percentage of the individuals with the highest fitness value to be taken over / cloned to the new generation. We use an elitism rate of 10 percent, i.e., the 50 best individuals will be taken without any changes of the genes to the new generation.

The population of each generation remains constant in time. In order to choose the remaining 450 individuals (parents) to generate the next generation, we use the *tournament selection* [30] including the implementation of the Apache Commons Mathematics Library. The tournament strategy can be configured by the *arity* of the tournament strategy, which specifies the number of individuals who take part in the tournament. For our purpose, five individuals in the tournament proved to be efficient. Accordingly, five individuals are selected randomly out of the total population of 500 individuals to take part in the tournament. Out of the individuals taking part in the tournament, the fittest individual is selected as a parent for the new generation. This way, 500 parents are selected out of a population of 500 individuals. These parents are paired aleatorically and they always have two descendants. This way, the next generation is created. Accordingly, the size of each generation remains constant.

We use two major strategies in order to improve the quality of the genes of the descendants, the *crossover strategy* described in [31] and the *mutation strategy*. Generally speaking, during the crossover phase, the two descendants receive the partly interchanged genes of their parents. Additionally, some particular genes can suffer mutations, i.e., the values of those particular genes are modified. The mutation policy can be configured, such that it allows or rejects the inclusion of new bins to the descendants. This way, the bin constraints can be reinforced, or the flexibility to mutate to any gene ensured. The general strategy to generate the descendants is based with some restrictions on random decisions.

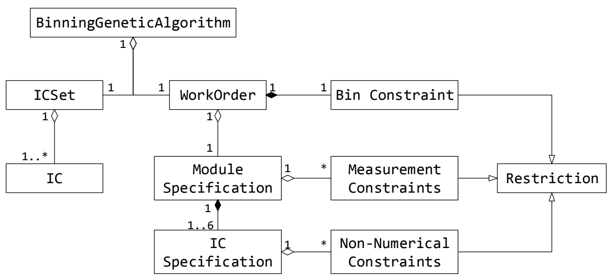


Figure 10: UML class diagram of algorithm and its parameters.

As shown in Figure 10, the genetic algorithm relies on the work order and the set of ICs as its principal input parameter. The ICs set holds the primary production result and the work order identifies the specification of the final product – i.e. set of modules –, which are delivered as the final product to the customer. The specification of the work order include the constraints of the IC, module and work order level. The constraints on the IC level were termed non-numerical constraints and relate to each particular IC, the constraints on module level were termed measurement constraints and they relate to the restriction the ICs that form the module have to satisfy as a group. Similar considerations apply for the bin constraints.

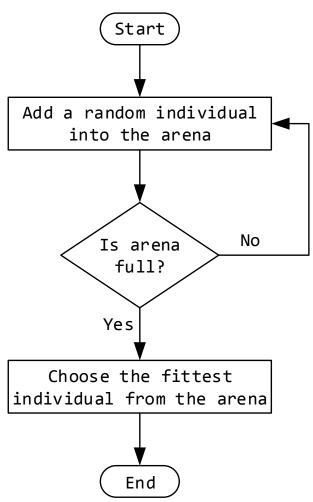


Figure 11: Tournament selection.

As mentioned above, according to the percentage given by the elitism rate, a subset of individuals of the current generation having promising genes are taken over unmodified to the new generation. In order to select the next pair of individuals during the process of establishing the new generation, see Figure 12, two groups of randomly chosen individuals are formed, the size of the group is configurable and given by the arity value. Then, the best individual from each arena is selected, circumventing the disadvantage of random selection. This method is depicted in Figure 11.

We describe in brief the creation strategy of the new generation. Some parameters are freely configurable, in order to assure best performance. Thus, the *crossover rate,* i.e., the threshold of the probability that a crossover is performed, has to be set in advance. Then, a crossover is performed if a randomly generated number is less than the crossover rate. Same is true regarding the mutation rate.

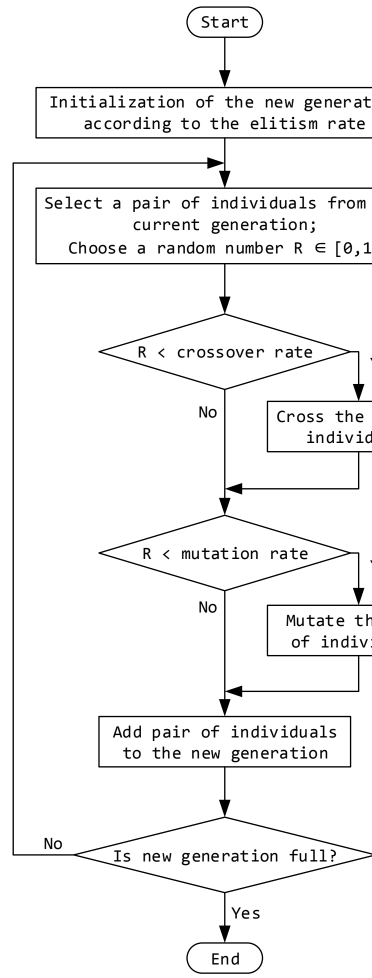


Figure 12: Example of the creation of a new generation.

The crossover policy is quite straightforward. The position and length of the genes to be crossed over are randomly generated and the two descendants have receive the interchanged genes of their parents. In our case, this policy has been improved, such that by in the end, the number of the bins of at least one descendant is using, is lower than (or if this is not possible equal to) the number of the bins of their parents. This way, the reduction of the number of bins an individual is using, is enforced by the crossover policy itself.

The mutation policy is also very intuitive. In addition to the mutation rate, which defines in the end, whether mutation is applied after the crossover phase or not, the exchange rate indicates whether a slot (IC) is to be renewed. Analogously, the mutation policy can be configured such that the number of bins the descendant is using is reduced or in worst case, kept constant, i.e., using the Bin Reduction Mutation (BRM) policy.

A simplified flowchart regarding the constitution of a new generation based on the current one is shown in Figure 12. First, based on the elitism rate, the best individuals of the current generation are cloned into the next generation. This way, the genes of the most promising individuals are saved for the next generation, this way, initializing the new generation. Next, either randomly or by tournament selection a new pair of individuals is selected from the current generation. According to a randomly generated number, crossover and mutation is performed on the pair of the individuals or on the single individuals, respectively. The pair of individuals generated this way is added to the set of individuals forming the new generation. The process of filling up the set of the new generation with new pairs is continued until the size of the new generation equals the size of the current generation.

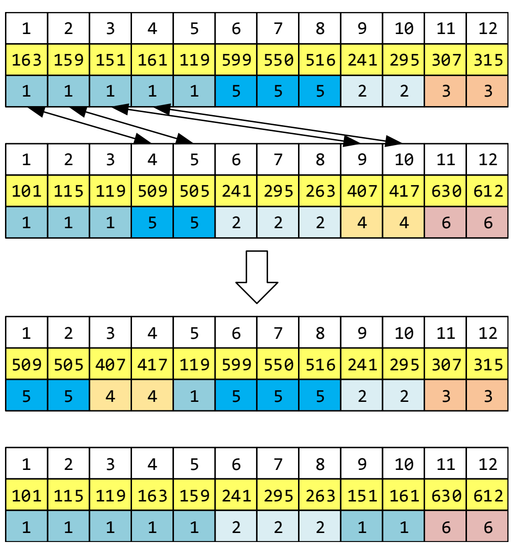


Figure 13: Example of bin reduction crossover policy.

Next, we illustrate based on a representative example in Figure 13, the Bin Reduction Crossover (BRC) policy. The BRC policy enhances the classical crossover policy, such that it uses the inside knowledge regarding the allocation of the ICs to the respective bins. As already mentioned, this allocation is fixed, it cannot be changed during the selection process. The general strategy of the BRC algorithm is to get rid of bins sparsely represented in exchange of ICs from much better represented bins. The two candidates dispose of ICs stored in some common bins, depicted in blue tones and and of unshared boxes, depicted in red tones. Some ICs can be found in both individuals, for example those with ID = 119, 241, 295. The basic idea is to move ICs from the common bins from one individual to another and in return to try to reduce the number of bins by returning ICs from sparsely bins. In our example, the ICs with ID = 136, 159, 151, 161 are moved from the first individual to the second, and the ICs with ID = 509, 505, 407, 417 are returned in order to balance the content of the individuals. This way, after the crossover is fulfilled, the second individual has genes from three boxes, which is a substantial reduction of bins used by the second individual. The overall number of bins composing the two individuals have been reduced this way by one, which corresponds to the envisaged objective.

An overview of establishing a new generation is depicted in Figure 14, which uses UML sequence for his flow chart. It uses implementation-related presentation and gives a glimpse of the Java code.

C. Performance Results

The benchmarks were performed on a Intel® Core^{TM} i5-6500 CPU (quad core CPU 3.2 GHz, 16 GB RAM) running on Windows 10 and Eclipse 3.7.0 using Java SE Runtime Environment 1.6.0_22. The genetic algorithm was implemented using the Apache Common Library, version 3.0. The test data is a subset of the production environment and contained 5518 ICs in 261 boxes, having 28 measurements on average. The restricted test data is a subset of the production environment and contained 27,590 ICs in 1,305 boxes. Due to the incomplete set of production data, only the two most critical modules are considered for selection. After taking into account the attributes corresponding to the specifications of the two modules regarding the ICs (article, number of pins, etc.) only eleven boxes contain ICs to be considered for the selection process. We term *pre-selection* the method to restrict the number of boxes by excluding those boxes which do not contain selectable elements. In this way, the search area can be drastically reduced and thus, the performance of the selection algorithm can be substantially improved.

We use a generation size of 500 individuals, an elitism rate of 10 percent and an arity value of 5. The number of generation is limited to 1000 and the runtime of the selection algorithm is limited to 300 seconds. The other parameters like the crossover rate and the mutation rate are configured on a case by case basis. Regarding the fitness function, the following configuration parameters have proved themselves as good choice: attribute weight = 1; measurement weight = 2; bin weight = 5. This means especially, that fulfilling the bin constraints is the most difficult one. In summary, we use the configuration parameters given in Table I for the performance tests.

TABLE I: SETTINGS OF CONFIGURATION PARAMETERS.

Configuration parameter	Value
Population size	500 individuals
Generation limit	1000 generations
Crossover policy	Bin reduction
Crossover rate	78 %
Mutation rate	13 %
Runtime limit	300 seconds
Elitism rate	10 %
Arity	5

The prediction of the results of the selection algorithm is hardly possible, since we use random strategies to generate the initial population, to select the parents for the next generation, to determine the crossover and mutation policy. Moreover, parameters like the elitism rate and the arity have to be configured. Hence, the interaction between many factors that

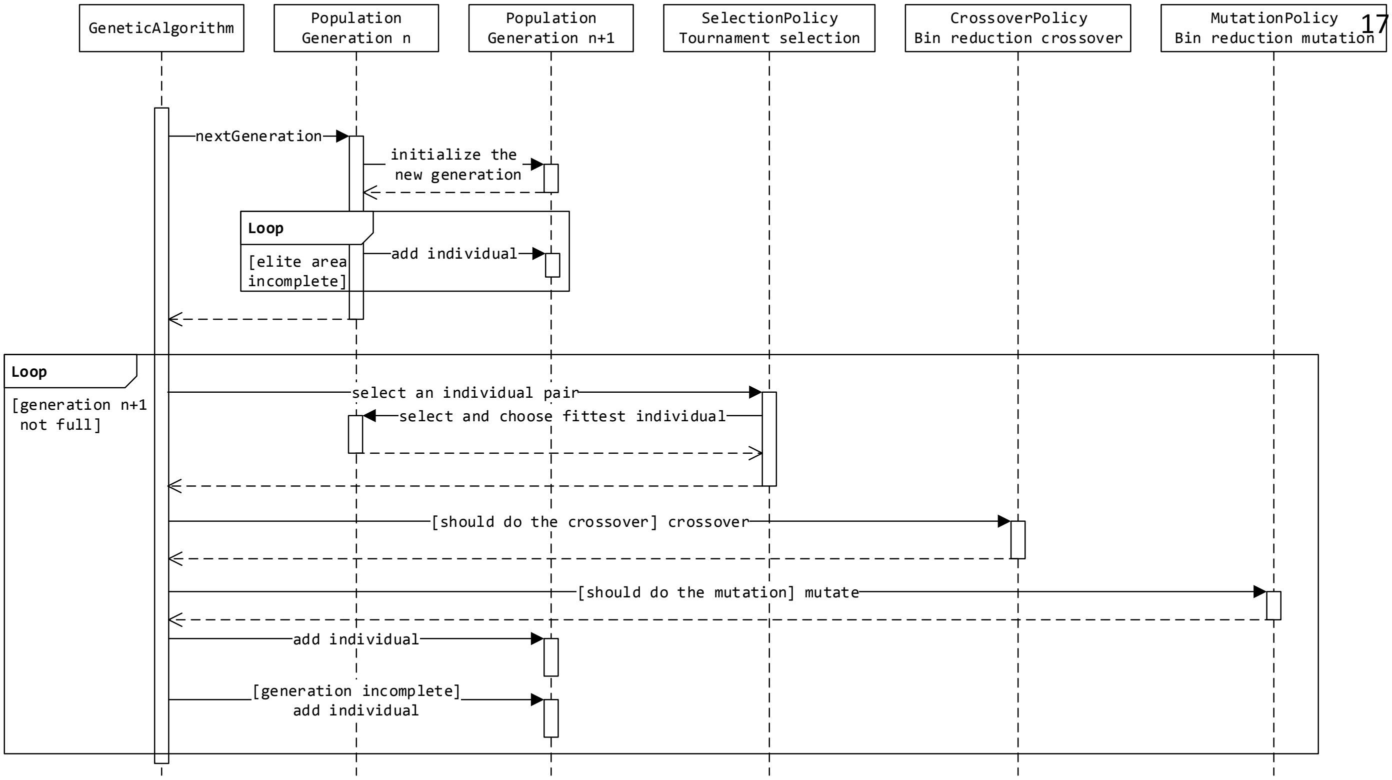


Figure 14: UML sequence diagram of a new generation.

can influence the success and performance of the selection algorithm is not obvious.

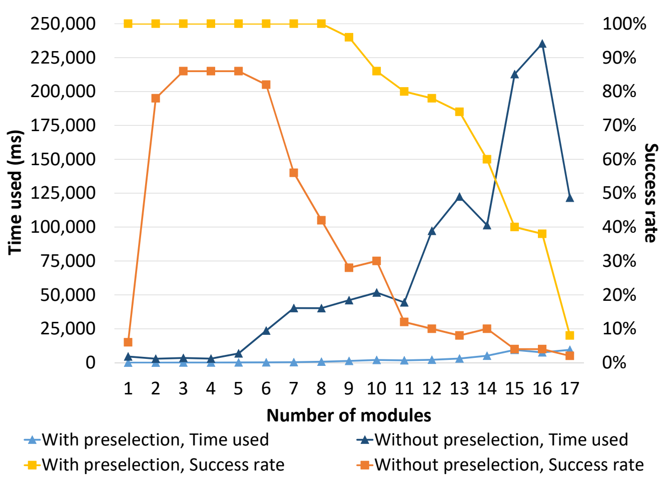


Figure 15: Apache: Success rate and time used depending on the number of modules with and without pre-selection.

It is not the aim of this study to deliver the possible best solution in an acceptable time frame and to improve the performance of the algorithm. Instead, our objective is to deliver an acceptable solution, i.e., a solution that fulfills the required constraints, for the industry to a crucial problem regarding their production problems. For example, there is no technological benefit of tightening the measurement constraints; the bin constraint was set up in such a way that seeking a lower value is not possible due to the fixed number of ICs of a work order and to the maximal capacity of the bins. Hence, the acceptable solution is also the best possible solution.

Nevertheless, we tried to improve the selection algorithm by testing the influence of the parameters, we find out to be decisive. This was also the case for the parameters of the fitness function as described above.

As already mentioned, we have an algorithm in place, which can find

a) a suboptimal solution in a heuristic way,

b) determine exactly whether the group of bins contain ICs which satisfies the specification of a particular work order.

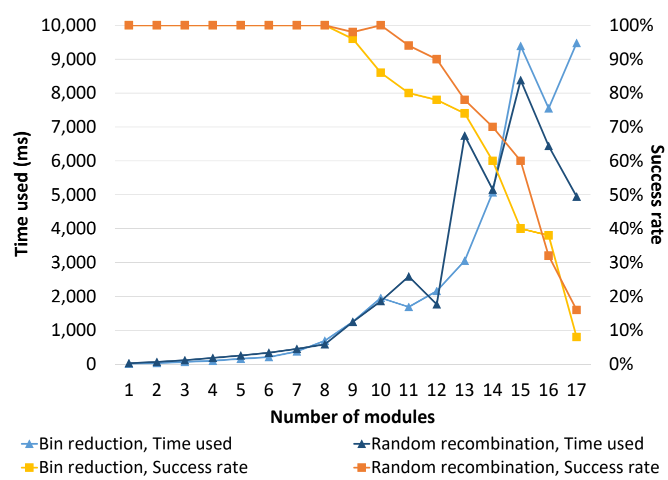


Figure 16: Apache: Success rate and time used when selecting the random recombination crossover policy or the bins reduction policy.

Figure 15 shows that the selection algorithm using pre-selection delivers the expected results, finding individuals having 19 modules. The *success rate,* i.e., the probability that the selection algorithm reaches with an individual the given number of modules, is over 60 percent and thus, high enough for practical systems. The pre-selection strategy is very straightforward and easy to implement. Thus, no practical system would renounce to it. Nevertheless, when neglecting the benefit of reducing the search space by using pre-selection, the results of the genetic algorithm are not always as promising as with pre-selection. In order to evaluate worst-case scenarios, we used work that posed a lot of difficulties to select with the heuristic algorithm in place. As illustrated in Figure 15, the success rate to select 19 modules as in the previous case, is at 60 percent. This means especially, that the successful run of the genetic algorithm heavily depends on the random numbers that were generated.

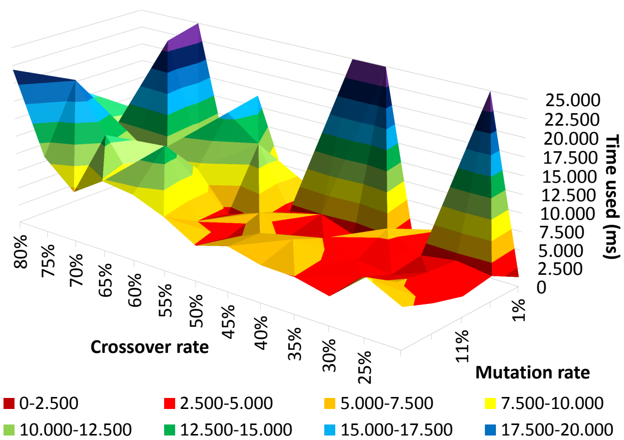


Figure 17: Apache: Time used depending on the crossover rate and the mutation rate.

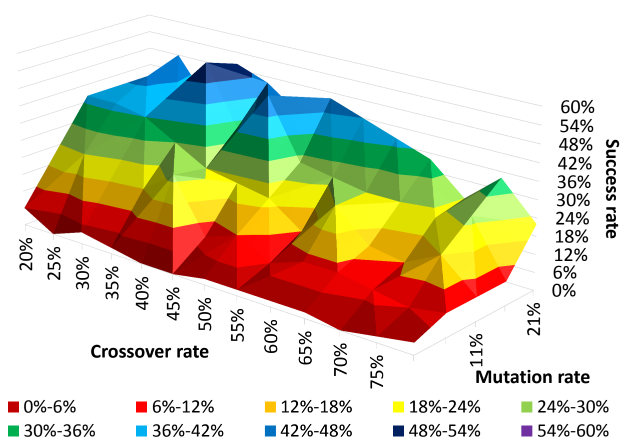


Figure 18: Apache: Success rate depending on the crossover rate and the mutation rate.

Figure 16 shows the difference between the random recombination crossover policy and the bins reduction policy. The boxes reduction crossover policy tries to reduce the number of boxes of the new individuals by focusing on the common bins of the parents. As a conclusion, using business logic over general approach, the general approach is as expected slower and has a lower success rate. This is the price to pay for using a more general solution over a customized one.

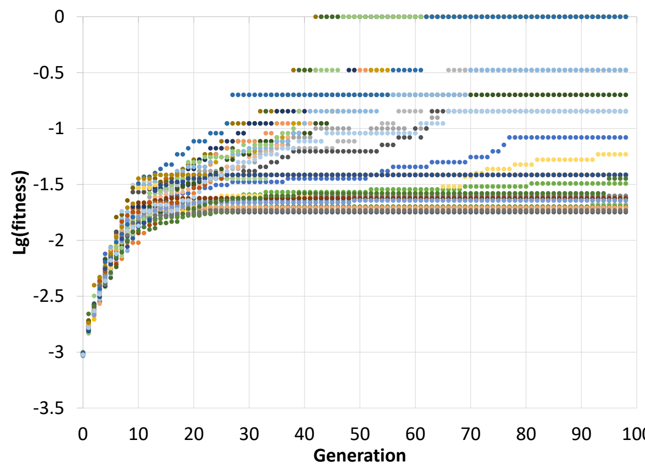


Figure 19: Logarithmic representation of the values of the fitness function depending on the number of generations (50 attempts).

Figures 17 and 18 show the influence of the crossover rate and the mutation rate to the success rate and the wall clock time. As not obvious at first glance, a smaller crossover rate and a higher mutation rate gives better values for the success rate. Keeping the crossover rate and the mutation rate low, better run time performance is achieved. Generally speaking, high mutation rate can destroy the structure of good chromosomes, if used randomly [32]. The above remark does not hold in our case, since we do not exchange ICs randomly, but according to our strategy to minimize the number of bins. We use for the three-dimensional graphics in this paper the best performing strategy, i.e. the bin reduction policy and pre-selection.

The tendency of the convergence of the fitness function is visualized in Figure 19. The graph shows that in the end all 50 threads converge after some generations, but only a subset to the envisaged value. Each dot indicates the best fitness value of a individual within one thread corresponding to a specific generation. Recall that the maximum value of the fitness function is per definition equal to 1, the higher the value of the fitness function, the better the solution. The values of the fitness functions are discrete, $\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, ...\}$. We use a decimal logarithmic representation, as a consequence the value 0 of $lg(fitness)$ indicates that a solution has been found. The best thread found a solution within 43 generations and the slowest thread - which is not represented in the graphic - found a solution after 228 generations. There are 8 threads in total which found a solution, the rest - although they converge - were not successful.

V. Resuming on Multi-Objective Optimization

Our preferred implementation framework is *Apache Commons Mathematics Library*, version 3.0 [33]. However, a very similar combinatorial grouping problem, the *Bin Packing Problem (BPP)* is investigated [34], by using the off the shelf *jMetal* framework [35]. The (one-dimensional) BPP [36] is defined as follows: given an unlimited number of bins with an integer capacity $c > 0$ each, a set of n items, $N = \{1, 2, ..., n\}$, and an integer weight $w_i, 0 < w_i \leq c$ for each item $i \in N$ assign each item to one bin, such that the total weight of the items in each bin does not exceed c and the number of bins used is minimized.

Luo et al. [34] extend the base implementation of jMetal to problems with dynamic variables. This was necessary, since the number of the genes in chromosomes is fixed in the base implementation of jMetal. However, the number of the genes in the specific implementation of the chromosomes for BPP – termed group based representation – is fluctuating; they vary in length depending on how many bins are used in every solution. Accordingly, the adopted implementation of BPP includes specific adaptations and enhancements of the basic primitives of jMetal, including those for chromosomes, crossover and mutation. The need for dynamic variables is justified by difficulties to use other solutions due to the fitness function.

In order to evaluate the performance of their algorithms – termed GABP –, Luo et al. [34] use public bench data as well as self-created big data sets. The performance of GABP does not differ very much from some of the known implementation of BPP. The main benefit of GABP is the implementation in a generic framework. However, the problem described in this article, the *Matching Lego(R)-Like Bricks Problem (MLBP)* is new to our knowledge, we are now aware of any implementation of a similar problem. The nearest problem to the MLBP seems to be BPP.

It seems that Luo et al. [34] used the fitness function as given below (termed cost function) [37] for their *group-based encoding scheme*:

$$f_{BPP} = \frac{1}{N_u} \cdot \sum_{i=1}^{N_u} (\frac{fill_i}{c})^k \tag{12}$$

with N_u being the number of bins used, $fill_i$ the sum of sizes of the objects in the bin i, c the bin capacity, and k a constant, $k > 1$. In other words, the cost function to maximize is the average over all bins used, of the k-th power of the bin's utilization of it's capacity. The authors state that experiments show that $k = 2$ gives good results. As Falkenauer and Delchambre [37] point out that one of the major purpose of the fitness function is to guide the algorithm in the search.

Although, both BPP and MLBP yield a reduction of the number of bins, the fitness function of the algorithms are different. The strategy at BPP is to pack the bins as full as possible, i.e., a bad use of the capacity of the bins leads to the necessity of supplementary bins [37]. On the contrary, the suggestion for the fitness function for MLBP is given by the need to fulfill the constraints.

By comparing the formula (2) for the fitness function of MLBP with the formula (12) as above, it is obvious that both formulas are very similar, both are expected to be maximized, contain summation over parameters of the respective problems and the possibility to optimize the execution time of the respective algorithms through clever setting of constants. In this respect, both MLBP and BPP substantially benefit from an ingeniously designed fitness function to ensure fast convergence towards the optimization goals [37].

Although, in concept MLBP is merely a constraint satisfaction problem, the implementation as described in this article, can be easily adapted to simulate an optimization problem. Within the current algorithm, the number of required bins is fixed. This assumption is perfectly reasonable from an industrial perspective, since the number of the objects in the bins is more or less the same and as the number of objects needed for a work order is known, the minimal number of bins necessary to fulfill the work order is thus determined. Besides, the maximum number of bins that can be accessed by a machine is fixed, less beans means just a vacant working place. However, by setting the bin constraints to a lower value, – while keeping the other constraints unchanged – and by modifying the exit criteria accordingly, the algorithm will loop – delivering better solutions according to the guidelines of the fitness function – until the new exit criteria are met. Thus, a local extremum relating to the fitness function is found. The local extremum is not automatically a solution to the constraint satisfaction problem, this has to be validated. The exit criteria only ensure that the best local value of the fitness function and corresponding physical entities are found. In this respect, the MLBP and BPP are closely related problems.

There are other systems, which provide frameworks of evolutionary algorithms, such as EvA2, OPT4j, ECJ, MOEAT, for a discussion and bibliography see [37]. Moreover, a remarkable attempt to obtain a deeper understanding of the structure of the BPP by using Principal Component Analysis and repercussions on the performance of the heuristic approaches to solve them, was undertaken [38].

The aim of jMetal was to set up a Java-based framework in order to develop meta heuristics for solving Multi-Objective Optimization Problems (MOOP). jMetal provides a rich set of Java classes as base components, which can be reused for the implementation of generic operators, thus making a comparison of different meta heuristics possible [39] [40].

Unfortunately, the Apache Commons Mathematics Library deployed in our use case, does not support multi-objective optimization mechanisms. This means especially, that multi-objective optimization have to be simulated by single-objective optimization. For example, optimization criteria for MLBP are a) reducing the number of bins, b) fulfillment of the measurement constraints, c) fulfillment of the attribute constraints. These criteria are independent of each other and ideally within the multi-objective optimization they can be optimized independently, such that an improvement of a criterion does not lead to a degradation of another one. For practical purposes, the fitness function, see formula (2), can be used for simulating multi-objective optimization by choosing adequate weight function. Thus, by choosing the values $w = 5$ for the criterion (a), $w = 2$ for criterion (b) and $w = 1$ for criterion (c) we achieve fast convergence and hence reduced execution time, but for example by improving criterion (a) we cannot avoid the degradation of criterion (b) or (c). By using frameworks which support multi-objective mechanism, better convergence of the genetic algorithm is expected.

The jMetal project was started in 2006 [39] and since then it underwent significant improvements and major releases [41], such that the redesigned jMetal should be useful to researchers of the multi-objective optimization community, such as evolutionary algorithm, evolution strategies, scatter search, particle swarm optimization, ant colony optimization, etc., [42]. Improvements regarding a new package for automatic tuning of algorithm parameter settings have been introduced [43] in order to facilitate accurate Pareto front approximations.

In addition, jMetal in conjunction with Spark – which is becoming a dominant technology in the Big Data context – have been used to solve Big Data Optimization problems by setting up a software platform. Accordingly, a dynamic bi-objective instance of the Traveling Salesman Problem based on near real-time traffic data from New York City has been solved [44] [45].

VI. Pursuing Considerations on Metaheuristics: An Excerpt

This section is a logical continuation of the last two preceding sections, it extents the approach of the use case in Section IV by considering the jMetal framework presented in Section V as the natural and fruitful terrain for further enhancement and advanced development.

A. General Considerations

We definitely do not intend to analyze in depth the Apache framework versus the jMetal framework; we restrict our analysis to the algorithm used in Section IV, which is ported to the jMetal framework. The comparison and test results should be interpreted accordingly, i.e., some other use case could lead to different conclusions. Moreover, the Apache and the jMetal frameworks are templates, which the developers can freely modify according to their needs, hence minor modifications in the templates could lead to unexpected major improvements.

```
//  gGA.java
//
//  Author:
//       Antonio J. Nebro <antonio@lcc.uma.es>
//       Juan J. Durillo <durillo@lcc.uma.es>
//
//  Copyright (c) 2011 Antonio J. Nebro, Juan J. Durillo
...
package jmetal.metaheuristics.singleObjective.
     geneticAlgorithm;
...
    while (evaluations < maxEvaluations) {
       ...
      // Copy the best two individuals to the offspring
           population
      offspringPopulation.add(new Solution(population.get
           (0))) ;
      offspringPopulation.add(new Solution(population.get
           (1))) ;
      // Reproductive cycle
         ...
        // Crossover; Mutation; Evaluation of the new
             individuals; Replacement: the two new
             individuals are inserted in the offspring
             population
        ...
      // The offspring population becomes the new current
           population
      ...
    } // while
...
```

Listing 1: Copy only the two best individuals to the offspring population.

The main objectives of this section are:

a) compare the multi bjective, jMetal framework based genetic algorithm versus the single-objective genetic, Apache framework based algorithm presented in Section IV,

b) compare the jMetal framework based genetic algorithm with other heuristic algorithms,

c) record the performance behavior of the investigated algorithms,

d) determine pros and cons of the Apache framework versus the jMetal framework and the different heuristic algorithms.

Unfortunately, the Apache framework does not support multi-objective optimization, hence in order to get around it, we switch to the jMetal framework, which has been deliberately developed to allow such comparisons. The best way to get directly familiar with jMetal or Apache in addition to studying the scientific literature [34] [35], [39]–[45] , is to read coding [46] [47].

Validation is the methodology to investigate whether individual bricks (ICs) or set of bricks satisfy the specified constraints. The validation fields – i.e., the structure that holds the information necessary to perform the decision regarding the creation of the next generation – is used both in the Apache and the jMetal framework based implementation of the genetic algorithm. The validation results are calculated during the validation process, i.e., during the process where it is checked whether the individual bricks or set of bricks fulfill the given constraints. This way, the overall quality of an individual during the reproduction process can be very well determined. As soon as the validation fields are fully filled, the information in this structure can be used to calculate the fitness value of the individuals and subsequently, it can be used to determine the unsuitable bricks and exchange them during the crossover and mutation process. As already presented, the fitness value contains the final score of an individual during the validation process, in other words, it represents the degree to which the individual is more closely to fulfill the specification of the work order. For example, if a brick breaks one of the constraints regarding the attribute values, it has to be removed during the mutation process, since the attributes are specific properties of each brick and unlike the measurement constraints, they are not influenced by the other bricks assigned to an individual.

```
// GeneticAlgorithm.java
// Licensed to the Apache Software Foundation (ASF) under
     one or more contributor license agreements.
...
package org.apache.commons.math3.genetics;
...
//Implementation of a genetic algorithm. All factors that
     govern the operation of the algorithm can be cond for
     a specific problem.
...
public class GeneticAlgorithm {
...
//Evolve the given population. Evolution stops when the
     stopping condition is satisfied.
...
        while (!condition.isSatisfied(current)) {
            current = nextGeneration(current);
            generationsEvolved++;
        }
...
}
```

Listing 2: Apache Commons Math. Stopping Condition

Summing up, the validation results of an individual contain in addition to the information regarding the final score, also details regarding the genes (bricks) which violate the constraints. This latter kind of information is necessary for further processing. Normally, the validation results of the parents should be determined before the mutation step is performed. In the Apache framework a pair of two individuals is provided and the crossover and mutation operations are performed directly on them. Unfortunately, this is not the case for the jMetal framework, it does not provides pairing, just individuals. As a consequence, since the validation fields are not part of the original framework, the validation results are not properly calculated and some – but not necessary all – of the mutations cannot be performed and are skipped. In order to overcome this anomaly, we modified the original implementation of the mutation procedure used in the Apache context, by adding an additional validation in order ensure that before mutation is performed, the parents dispose of the appropriate validation results.

Unfortunately, jMetal has only one predefined stopping condition, i.e., the algorithm can stop only if the number of evaluations, i.e., in our case the number of generations surpasses some threshold – i.e., *maxEvaluations* – compare Listing 1, line 11. In contrast, the Apache framework allows additional to the condition above, multiple and flexible stopping conditions:

a) exit after some predefined time,

b) exit if the envisaged result has been found, etc., see Listing 2, line 12 for an example,

c) exit according to any user defined conditions, due to the flexibility of the Apache environment.

```
// ElitisticListPopulation.java
// Licensed to the Apache Software Foundation (ASF) under
     one or more contributor license agreements.
 ...
package org.apache.commons.math3.genetics;
...
//Population of chromosomes which uses elitism (certain
     percentage of the best chromosomes is directly copied
     to the next generation).
 ...
public class ElitisticListPopulation extends
     ListPopulation {
    // percentage of chromosomes copied to the next
         generation
    private double elitismRate =10.0;
...
    // Creates a new ElitisticListPopulation instance.
     ...
     // Start the population for the next generation.
     ...
    public Population nextGeneration() {
        // initialize a new generation with the same
             parameters
...
        // index of the last "not good enough" chromosome
        for (int i=boundIndex; i<oldChromosomes.size(); i
             ++) {
            nextGeneration.addChromosome(oldChromosomes.
                 get(i));
        }
...
}
```

Listing 3: Apache Commons Math. Elitism Rate

The jMetal framework also uses a predefined type of population, characterized by the elitism rate, which specifies the percentage of individuals with the highest fitness value to be taken over unmodified to the new generation. This way, the genetic algorithm will put directly some of his best solutions unmodified to the next generation. This strategy has been successfully used with the Apache framework to optimize the run time of the genetic algorithm. Unfortunately, the elitism rate is hard coded and cannot be configured in the jMetal framework. Only the two best solutions are forwarded unmodified to the next generation, compare Listing 1, line 17,18. This means by considering our population size of 500 individuals, an elitism rate of 0.4%. We obtained best performance results within the Apache framework with this rate set to 10 in the benchmarks of Section IV, see Listing 3. In contrast to the jMetal framework, where it is not possible to create or define a new population type, the Apache framework can use a lot of other population types apart from the class *ElitisticListPopulation*, for example population types based on random selection of the individuals or based on keeping diversity selection. These new population types can be set up either by the user itself, or the development team of the Apache framework. Of course, random selection can be a promising approach, since some of the best genes can be hidden in bad individuals and those individuals might evolve later. The strategy to diversify means especially, that no similar individuals will be directly selected for the next generation. Obviously, the concept of similarity has to be specified accordingly, for example, by trying to have as much bins involved in the generation of the initial population, we avoid to have a restricted set of bins in the initial population that has no solution at all.

As mentioned above, the jMetal framework is less flexible than the Apache framework, i.e., the possibility to configure the jMetal framework is less pronounced. Hence, we cannot compare the performance of those frameworks directly, instead we are obliged to switch to those less efficient configuration parameters that work for both frameworks in order to be able to compare similar runs. We intend to do performance benchmarks as follows:

a) Apache versus jMetal (Apache and jMetal best performing configuration);

b) Apache versus jMetal (Apache and jMetal same configuration);

c) jMetal single objective genetic algorithm versus jMetal multi-objective genetic algorithm;

d) jMetal genetic algorithm versus jMetal other heuristic algorithms.

B. Used Methods and Strategies

Evolutionary algorithms such as the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [28], [43], [48] and Strength Pareto Evolutionary Algorithm 2 (SPEA-2) [49], [50] have become standard approaches, although some approaches based on Particle Swarm Optimization (PSO) [51], simulated annealing, and Scatter Search (SS) [52] are best known. Durillo [39] gives a very good overview of jMetal, describing some of the evolutionary algorithms implemented, for example NSGA-II, SPEA-2, Pareto Archived Evolution Strategy (PAES), Optimal multi-objective Optimization based on Particle Swarm Optimization (OMOPSO) [53], Archive based hYbrid Scatter Search (AbYSS), etc. The PSO algorithm has been used successfully not only for continuous optimization problems, but also to a combinatorial optimization of a real university course timetabling problem [54], see [55] for a more recent study regarding non-continuous search spaces.

Besides jMetal there are further frameworks for meta-heuristic optimization, for example Opt4J [56]. This framework is modular, i.e., it supports the optimization of complex optimization tasks by allowing its decomposition into subtasks that may be designed and developed separately. Additional optimization frameworks, java-based and non-java-based are listed on the SourceForge site [57], where also the latest release can be found. Additionally, a framework that works as a top-level layer over generic optimization frameworks that implement a high number of metaheuristics proposed in the technical literature, such as jMetal and Opt4J is presented in [58]. JECoLi, another novel Java-based library for the implementation of metaheuristic optimization algorithms with a focus on Genetic and Evolutionary Computation based methods is introduced [59].

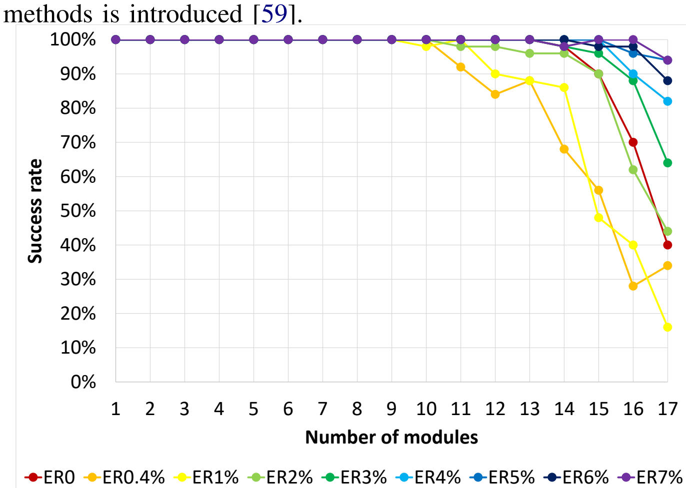


Figure 20: Success rate depending on the elitism rate.

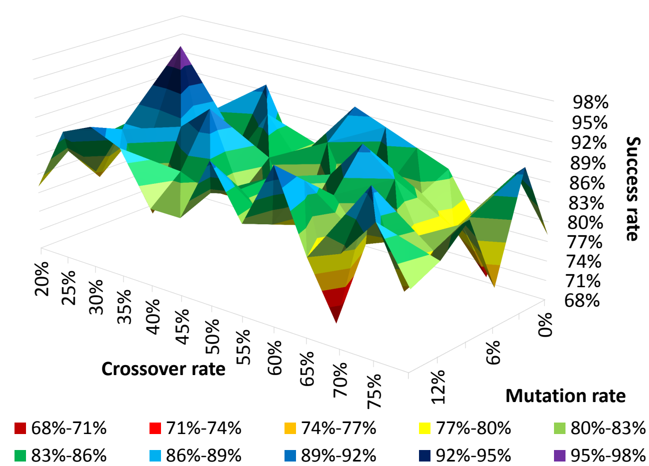


Figure 21: jMetal single-objective optimization: Success rate depending on the crossover rate and the elitism rate.

C. Performance Results

This subsection is aligned with Subsection IV-C as far as the testing strategy is concerned, such that performance and accuracy comparison are possible. Accordingly, the test scenarios in this subsection are similar to those in Figure 15, Figure 16, Figure 17, Figure 18 as done for the Apache Framework. The configuration parameter for the test settings are as in Table I.

1) jMetal Single-Objective Genetic Algorithm: Actually, the naive expectation regarding the similitude between the performance behavior of the Apache Framework and the single-objective optimization part of the jMetal Framework has not been met. Furthermore, for the Apache Framework, the elitism rate is not so sensitive regarding the fluctuation of the success rate, whereas for the jMetal Framework this is true for the mutation rate.

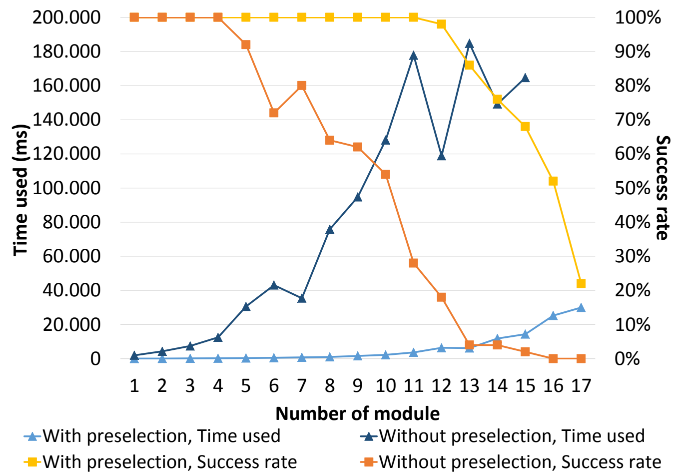


Figure 22: jMetal single-objective optimization: Success rate and time used depending on the number of modules with and without pre-selection.

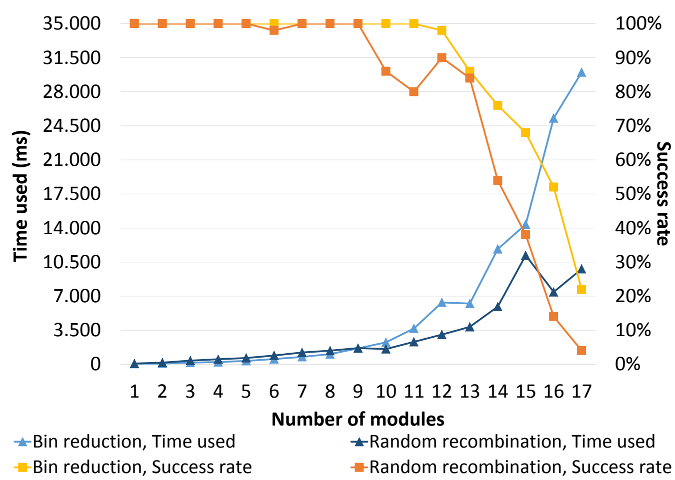


Figure 23: jMetal single-objective optimization: Success rate and time used when selecting the random recombination crossover policy or the bins reduction policy.

Figure 22 corresponds to Figure 15 and Figure 23 corresponds to Figure 16 respectively, generated for genetic algorithm using the Apache Framework. For a detailed background description see the corresponding explanation in Subsection IV-C. Figure 24 corresponds to Figure 17 and Figure 25 corresponds to Figure 18 respectively, generated for genetic algorithm using the Apache Framework.

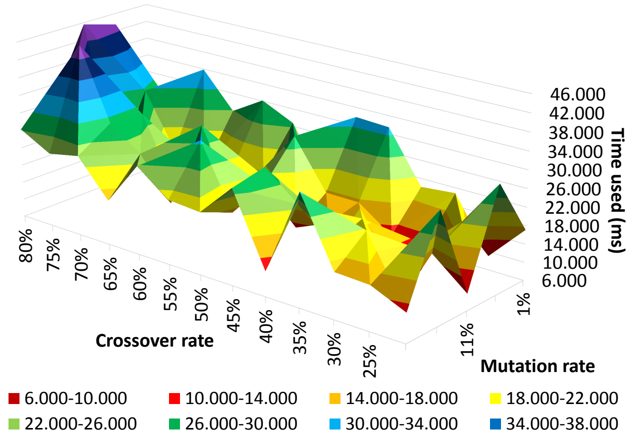


Figure 24: jMetal single-objective optimization: Time used depending on the crossover rate and the mutation rate.

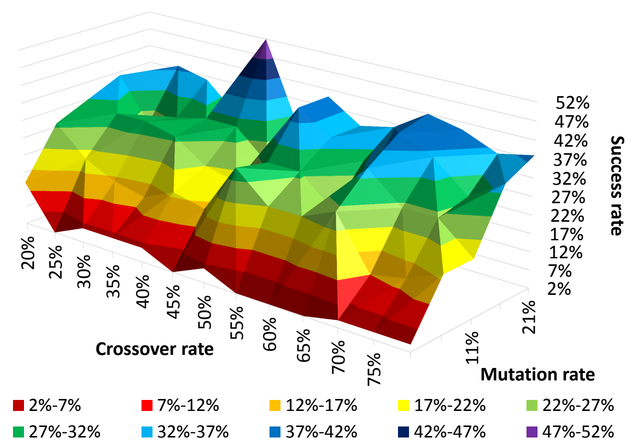


Figure 25: jMetal single-objective optimization: Success rate depending on the crossover rate and the mutation rate.

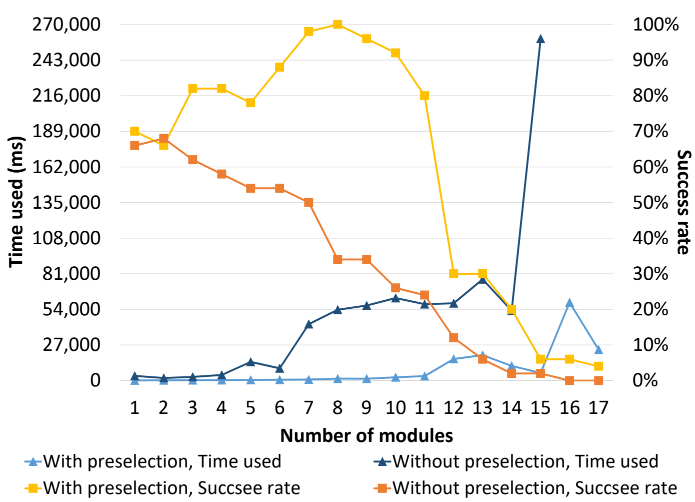


Figure 26: jMetal multi-objective optimization using NSGA-II : Success rate and time used depending on the number of modules with and without pre-selection.

Unexpectedly, the elitism rate has a major influence on the success rate, even for lower values, see Figure 20 for a test scenario with the elitism rate $\in [0\%, 7\%]$. The other configuration parameters for the performance tests are given in Table I. The optimal value for the elitism rate is around $\%7$, unfortunately the elitism rate is not configurable in the official release of jMetal. As already mentioned, to circumvent this shortcoming, we have adapted the original open source code accordingly. As is Figure 21, increasing the elitism rate does not improve the success rate, moreover an appropriate tuple of crossover rate and elitism rate seems to be the most effective.

For a detailed description of the corresponding performance results, see the explanation in Subsection IV-C. A low crossover rate and low elitism rate gives the success rate at over 95%. Increased crossover rates and increase elitism rates also deliver success rates at over 90%. Unfortunately, high success rates also deteriorates considerably its run time aspects.

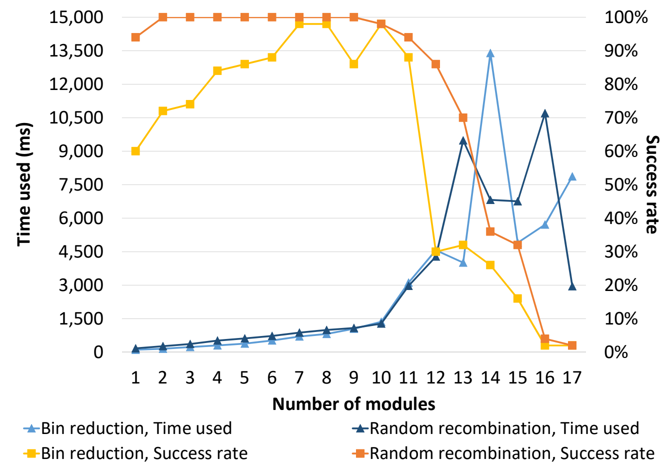


Figure 27: jMetal multi-objective optimization using NSGA-II : Success rate and time used when selecting the random recombination crossover policy or the bins reduction policy.

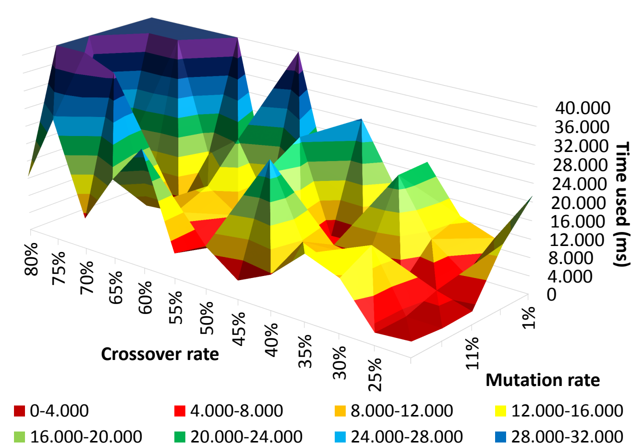


Figure 28: jMetal multi-objective optimization using NSGA-II : Time used depending on the crossover rate and the mutation rate.

2) jMetal multi-objective Genetic Algorithm: Unfortunately, NSGA-II does not use the concept of elitism rate to select offsprings, it uses Pareto front and distance instead. Hence we cannot compare directly the similar tests for single-objective optimization based on crossover rate and elitism rate, see Figure

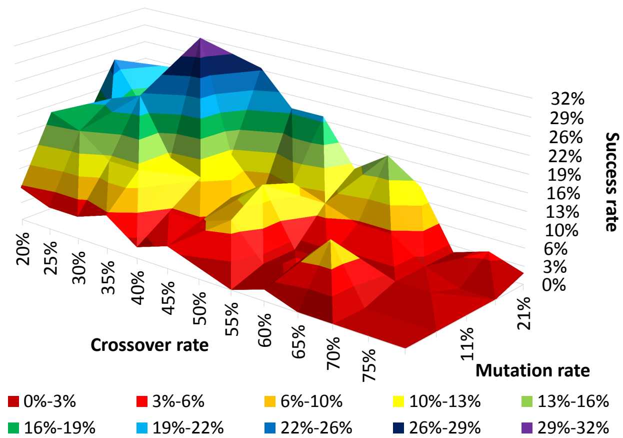


Figure 29: jMetal multi-objective optimization using NSGA-II : Success rate depending on the crossover rate and the mutation rate.

Figure 26 corresponds to Figure 15 and Figure 27 corresponds to Figure 16 respectively, generated for genetic algorithm using the Apache Framework. For a detailed background description see the corresponding explanation in Subsection IV-C. Figure 28 corresponds to the Figure 17 and Figure 29 corresponds to Figure 18 respectively, generated for the genetic algorithm using the Apache Framework.

3) Other evolutionary algorithms: To conclude, we examine the scatter search template adapted to a hybrid - single objective optimization to the multi-objective domain - called Archive-Based hYbrid Scatter Search (AbYSS) [60]–[63]. AbYSS follows the scatter search structure, but uses mutation and crossover operators from evolutionary algorithms. It incorporates typical concepts from the multi-objective field, such as Pareto dominance, etc.

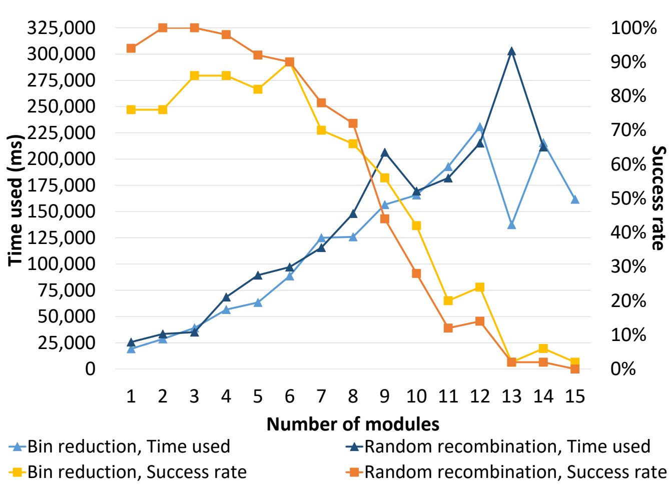


Figure 30: jMetal AbYSS hybrid optimization : Success rate and time used when selecting the random recombination crossover policy or the bins reduction policy.

We generate appropriate tests – as represented in Figure 30, see Figure 16 for the Apache counterpart – concerning the success rate and time used by using the random recombination crossover policy and the bins reduction policy respectively. According to [61] the results of their tests show that AbYSS not only reaches very accurate solutions, but also it scales well with increasingly sized instances. Unfortunately, the performance and accuracy of our tests is lower than expected, such that the AbYSS algorithm finds no more than 15 modules, whereas the number of possible solution is 17, see Figure 30 for the Apache Framework genetic algorithm counterpart. We cease further testing.

4) Final conclusions: Summing up, low crossover and low elitism rates or high crossover and high elitism rates deliver good results for the success rate. On the contrary, high success rates are always achieved with high timed used rates. The success rate is not very sensitive regarding the mutation rates as long as they are low or moderate. Satisfactory results have been achieved with elitism rates exceeding 3%. It seems that the higher success rate in the single-objective optimization case is due to a very well balanced fitness function, whereas if no time for performance improvement tests, the multi-objective path delivers satisfactory results with lower implementation effort. The run time performance of our application implemented using the Apache Framework genetic algorithm is order of magnitudes better than the corresponding implementation based on the jMetal Framework.

VII. Conclusion and Future Work

The main challenge, which led to the results of this paper, was to investigate whether a real-life combinatorial problem, which popped up at a semiconductor company, can be solved within a reasonable time. Moreover, the solution should be flexible, such that the solution is not restricted to the existing specification of the modules.

We established an abstract formal model, such that the implementation of the use case is fully functional within the boundaries of this model. In this sense, new constraints can be added to the existing ones, no inside knowledge regarding the structure of the module is needed, as it is the case for the heuristic algorithm in place.

We set up a genetic algorithmic approach based on the Apache Commons Mathematics Library, implemented it and validated the results. Some decisive policies like the crossover and mutation policy have been additionally implemented and new optimizations like the bin reduction crossover policy have been set up to improve the convergence of the genetic algorithm. The performance results were satisfactory for an industrial application.

The current implementation does not combine good genes (taking into account all the constraints, like measurement, etc.) of the parents. Instead, the crossover strategy is based on random decisions. Additional research is necessary in this direction to find a good balance between a more general suitability (random decisions) and good convergence (adjusted crossover policy). For the time being, only the fitness function contains proprietary information regarding the production process, any other decision is aleatoric.

However, improvements such as the bin reduction crossover policy and improvement of the bin preserving mutation policy based on internal knowledge have in most cases advantageous effects on the selection algorithm. Unfortunately, the advantages of these improvements, like run time reduction and improved accuracy, are achieved by less flexibility. Whenever the external circumstances change, the corresponding policies have to be adapted accordingly.

The implemented basic framework is very flexible, it has many configuration possibilities like the elitism rate and the arity. As a consequence of the random variables, many convergence tests with various configuration assignment have to be performed, in order to ensure satisfying results. Furthermore, of crucial importance for the successful completion of the algorithm is the design of the fitness function, especially the values of the weights.

The convergence tests show that not every execution will succeed to find the best solution delivered, this is due to the random numbers used through out the genetic algorithm. This is exemplified by the success rate. Thus, the selection algorithm based on genetic strategies always delivers local maxima, which may substantially differ from the global one. Our attempt to find the optimal solution using MATLAB for a reduced set of ICs failed due to the long execution time.

As already mentioned, the fitness function plays an outstanding role during the selection of the best candidates for the next generation. This means especially, that two candidates having the same value or fitness function are considered of the same quality. This assertion is not accurate enough, due to the use of three different weights, whose interdependence can hardly be anticipated. Each weight represents the significance of one aspect of the quality of a candidate. To circumvent this dilemma, Pareto optimality [29] can be used to solve the challenge of the multi-objective function. In this case, a new framework [39] is needed, since Apache Commons Mathematics Library does not support multi-objective mechanism. Genetic algorithms, if configured properly, can be used to solve our constraint satisfaction problem. The delivered solution may substantially differ from the optimal one.

The current problem is not defined as an optimization problem, the constraints of a work order are either satisfied or not. Accordingly, two different solutions of the same work order, which satisfy the constraints, are of the same quality. However, the genetic algorithm is based internally on an optimization process – the higher the value of the fitness function, the better the solution. The constraints are used as exit criterion for the genetic algorithm. In this way, the optimization is stopped arbitrarily, considering that a better solution is out of scope.

Moreover, during the production process multiple work orders have to be honored simultaneously. The current strategy at the semiconductor company adopted a sequential one. We can reformulate the problem as an optimization problem: Given a list of work orders, find the maximum number of work orders that can be satisfied simultaneously.

The elapsed time till the genetic algorithm of MLBP converges is in range of seconds, by all means satisfactory for the investigated industrial application. As expected, not all the execution threads converge to the same solution, and not all the threads find an optimal solution, as shown in some cases less than 60 percent. Therefore, starting a bunch of threads within the genetic algorithm increases the chance towards better solutions.

To summarize, regarding the configuration possibilities, the jMetal framework is less flexible than the Apache framework, i.e., the Apache framework allows much more alternatives for settings, in contrast in order to achieve greater adjustability in the jMetal framework, the source code has to be adapted accordingly. The jMetal framework predefines a lot of heuristic algorithms, but they have to be refined and improved in order to reach the profoundness of the Apache framework.

ACKNOWLEDGMENT

We acknowledge the assistance and helpful comments of the anonymous referees.

REFERENCES

[1] M. Zinner, K. Feldhoff, R. Song, A. Gellrich, and W. E. Nagel, "The Matching Lego(R)-Like Bricks Problem: Including a Use Case Study in the Manufacturing Industry," ICSEA 2019, The Fourteenth International Conference on Software Engineering Advances, 2019, pp. 130–140, Retrieved: December 2020. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=icsea_2019_6_10_10107

[2] B. Korte and J. Vygen, Combinatorial Optimization: Theory and Algorithms, ser. Algorithms and Combinatorics. Springer Berlin Heidelberg, 2018.

[3] P. M. Pardalos, D.-Z. Du, and R. L. Graham, Handbook of Combinatorial Optimization. Springer, 2013.

[4] J. Puchinger and G. Raidl, "Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification," Lecture Notes in Computer Science, vol. 3562, 06 2005, pp. 41–53, doi: 10.1007/11499305_5.

[5] Krzysztof Apt, Principles of Constraint Programming. Cambridge University Press, 2003, Retrieved: December 2020. [Online]. Available: https://doi.org/10.1017/CBO9780511615320

[6] A. L. Corcoran, Using LibGA to Develop Genetic Algorithms for Solving Combinatorial Optimization Problems. The Application Handbook of Genetic Algorithms, Volume I, Lance Chambers, editor, pages 143-172 CRC Press, 1995.

[7] E. Andrés-Pérez et al., Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems. Springer, 2018, vol. 49.

[8] D. Greiner et al., Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences. Springer, 2015, vol. 1, no. 1.

[9] D. Simon, Evolutionary Optimization Algorithms. John Wiley & Sons, 2013.

[10] A. Pétrowski and S. Ben-Hamida, Evolutionary Algorithms. John Wiley & Sons, 2017.

[11] O. Kramer, Genetic Algorithm Essentials. Springer, 2017, vol. 679.

[12] T. Bäck, D. B. Fogel, and Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators. CRC press, 2018.

[13] R. K. Belew, Adaptive Individuals in Evolving Populations: Models and Algorithms. Routledge, 2018.

[14] T. Bäck, D. B. Fogel, and Z. Michalewicz, Evolutionary Computation 2 : Advanced Algorithms and Operators. CRC Press, 2000, Textbook - 308 Pages.

[15] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic Research: a Comprehensive Survey," Artificial Intelligence Review, 2018, pp. 1–43.

[16] T. Jiang and C. Zhang, "Application of Grey Wolf Optimization for Solving Combinatorial Problems: Job Shop and Flexible Job Shop Scheduling Cases," IEEE Access, vol. 6, 2018, pp. 26 231–26 240.

[17] H. Zhang, Y. Liu, and J. Zhou, "Balanced-Evolution Genetic Algorithm for Combinatorial Optimization Problems: the General Outline and Implementation of Balanced-Evolution Strategy Based on Linear Diversity Index," Natural Computing, vol. 17, no. 3, 2018, pp. 611–639.

[18] X. Li, L. Gao, Q. Pan, L. Wan, and K.-M. Chao, "An Effective Hybrid Genetic Algorithm and Variable Neighborhood Search for Integrated Process Planning and Scheduling in a Packaging Machine Workshop," IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018.

[19] I. Jackson, J. Tolujevs, and T. Reggelin, "The Combination of Discrete-Event Simulation and Genetic Algorithm for Solving the Stochastic Multi-Product Inventory Optimization Problem," Transport and Telecommunication Journal, vol. 19, no. 3, 2018, pp. 233–243.

[20] J. C. Bansal, P. K. Singh, and N. R. Pal, Evolutionary and Swarm Intelligence Algorithms. Springer, 2019.

[21] G. Raidl, J. Puchinger, and C. Blum, "Metaheuristic hybrids," Handbook of Metaheuristics, Vol. 146 of International Series in Operations Research and Management Science, 09 2010, pp. 469–496, doi: 10.1007/978-1-4419-1665-5_16.

[22] J. Holland, Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor. 2nd Edition, MIT Press, 1992.

[23] G. P. Rajappa, "Solving Combinatorial Optimization Problems Using Genetic Algorithms and Ant Colony Optimization," 2012, PhD diss., University of Tennessee, Retrieved: December 2020. [Online]. Available: https://trace.tennessee.edu/utk_graddiss/1478

[24] S. Yakovlev, O. Kartashov, and O. Pichugina, "Optimization on Combinatorial Configurations Using Genetic Algorithms," in CMIS, 2019, pp. 28–40.

[25] T. Weise, "Global Optimization Algorithms – Theory and Application," 2009, Retrieved: December 2020. [Online]. Available: http://www.it-weise.de/projects/book.pdf

[26] H. P. Christos and K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity," Prentice Hall Inc., 1982.

[27] Rui Song, "Substrate Binning for Semiconductor Manufacturing," Master's thesis, Dresden University of Technology, Faculty of Computer Science, Institute of Applied Computer Science, Chair of Technical Information Systems, 2013.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, no. 2, 2002, pp. 182–197.

[29] J. rey Horn, N. Nafpliotis, and D. E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," in Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence, vol. 1. Citeseer, 1994, pp. 82–87.

[30] B. L. Miller, D. E. Goldberg et al., "Genetic Algorithms, Tournament Selection, and the Effects of Noise," Complex systems, vol. 9, no. 3, Champaign, IL, USA: Complex Systems Publications, Inc., 1995, pp. 193–212.

[31] R. Poli and W. B. Langdon, "A New Schema Theorem for Genetic Programming with One-Point Crossover and Point Mutation," Cognitive Science Research Papers-University of Birmingham CSRP, 1997.

[32] F.-T. Lin, "Evolutionary Computation Part 2: Genetic Algorithms and Their Three Applications," Journal of Taiwan Intelligent Technologies and Applied Statistics, vol. 3, no. 1, 2005, pp. 29–56.

[33] M. Andersen et al., "Commons Math: The Apache Commons Mathematics Library," URL http://commons. apache. org/math/, online, 2011, Retrieved: December 2020.

[34] F. Luo, I. D. Scherson, and J. Fuentes, "A Novel Genetic Algorithm for Bin Packing Problem in jMetal," in 2017 IEEE International Conference on Cognitive Computing (ICCC). IEEE, 2017, pp. 17–23.

[35] J. J. Durillo and A. J. Nebro, "jMetal: A Java Framework for Multi-Objective Optimization," Advances in Engineering Software, vol. 42, no. 10, Elsevier, 2011, pp. 760–771.

[36] K. Fleszar and C. Charalambous, "Average-Weight-Controlled Bin-Oriented Heuristics for the One-Dimensional Bin-Packing Problem," European Journal of Operational Research, vol. 210, no. 2, Elsevier, 2011, pp. 176–184.

[37] E. Falkenauer and A. Delchambre, "A Genetic Algorithm for Bin Packing and Line Balancing," in Proceedings 1992 IEEE International Conference on Robotics and Automation. IEEE, 1992, pp. 1186–1192.

[38] E. López-Camacho, H. Terashima-Marín, G. Ochoa, and S. E. Conant-Pablos, "Understanding the Structure of Bin Packing Problems Through Principal Component Analysis," International Journal of Production Economics, vol. 145, no. 2, Elsevier, 2013, pp. 488–499.

[39] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba, "jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics," Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Tech. Rep. ITI-2006-10, 2006.

[40] A. J. Nebro and J. J. Durillo, "jMetal 4.4 User Manual," Available from Computer Science Department of the University of Malaga, 2013, Retrieved: December 2020. [Online]. Available: http://jmetal.sourceforge.net/resources/jMetalUserManual44.pdf

[41] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal Multi-Objective Optimization Framework," in Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation. ACM, 2015, pp. 1093–1100.

[42] J. J. Durillo, A. J. Nebro, and E. Alba, "The jMetal Framework for Multi-Objective Optimization: Design and Architecture," in IEEE Congress on Evolutionary Computation. IEEE, 2010, pp. 1–8.

[43] A. J. Nebro-Urbaneja et al., "Automatic Configuration of NSGA-II with jMetal and irace," ACM, 2019.

[44] J. A. Cordero et al., "Dynamic Multi-Objective Optimization with jMetal and Spark: a Case Study," in International Workshop on Machine Learning, Optimization, and Big Data. Springer, 2016, pp. 106–117.

[45] C. Barba-González, J. García-Nieto, A. J. Nebro, and J. F. A. Montes, "Multi-objective Big Data Optimization with jMetal and Spark," in EMO, 2017.

[46] Apache Commons Math, "Genetic algorithms," 2016, Home Page, Retrieved: December 2020. [Online]. Available: https://commons.apache.org/proper/commons-math/userguide/genetics.html

[47] SOURCEFORGE, "jmetal," 2017, Home Page, Retrieved: December 2020. [Online]. Available: https://sourceforge.net/projects/jmetal/

[48] A. J. Nebro, J. J. Durillo, C. A. C. Coello, F. Luna, and E. Alba, "A Study of Convergence Speed in Multi-Objective Metaheuristics," in International Conference on Parallel Problem Solving from Nature. Springer, 2008, pp. 763–772.

[49] Z. Jalali, E. Noorzai, and S. Heidari, "Design and Optimization of Form and Façade of an Office Building Using the Genetic Algorithm," Science and Technology for the Built Environment, vol. 26, no. 2, 2020, pp. 128–140.

[50] I. Miri, M. H. Fotros, and A. Miri, "Comparison of Portfolio Optimization for Investors at Different Levels of Investors' Risk Aversion in Tehran Stock Exchange with Meta-Heuristic Algorithms," Advances in Mathematical Finance and Applications, vol. 5, no. 1, 2020, pp. 1–12.

[51] T. Hendtlass, "The Particle Swarm Algorithm," in Computational Intelligence: A Compendium. Springer, 2008, pp. 1029–1062.

[52] R. Ba C. Gil, J. Reca, and J. Martz, "Implementation of Scatter Search for Multi-Objective Optimization: A Comparative Study," Computational Optimization and Applications, vol. 42, 04 2009, pp. 421–441.

[53] M. R. Sierra and C. A. C. Coello, "Improving PSO-based Multi-Objective Optimization Using Crowding, Mutation and?-Dominance," in International conference on evolutionary multi-criterion optimization. Springer, 2005, pp. 505–519.

[54] O. Brodersen and M. Schumann, "Einsatz der Particle Swarm Optimization zur Optimierung universitärer Stundenpläne," Techn. Rep, vol. 5, 2007.

[55] V. Santucci, M. Baioletti, and A. Milani, "Tackling Permutation-Based Optimization Problems with an Algebraic Particle Swarm Optimization Algorithm," Fundamenta Informaticae, vol. 167, no. 1-2, 2019, pp. 133–158.

[56] Lukasiewycz, Martin and Glaß, Michael and Reimann, Felix and Teich, Jürgen, "Opt4J: A Modular Framework for Meta-Heuristic Optimization," in Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011, pp. 1723–1730.

[57] SOURCEFORGE, "Opt4j," 2017, Home Page, Retrieved: December 2020. [Online]. Available: http://opt4j.sourceforge.net/

[58] A. D. S. Grande, R. D. F. Rodrigues, and A. C. Dias-Neto, "A Framework to Support the Selection of Software Technologies by Search-Based Strategy," in 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. IEEE, 2014, pp. 979–983.

[59] P. Evangelista, P. Maia, and M. Rocha, "Implementing Metaheuristic Optimization Algorithms with JECoLi," in 2009 Ninth International

Conference on Intelligent Systems Design and Applications, 2009, pp. 505–510.

[60] A. J. Nebro et al., "AbYSS: Adapting Scatter Search to Multiobjective Optimization," IEEE Transactions on Evolutionary Computation, vol. 12, no. 4, 2008, pp. 439–457.

[61] F. Luna, J. J. Durillo, A. J. Nebro, and E. Alba, "A Scatter Search Approach for Solving the Automatic Cell Planning Problem," in Large-Scale Scientific Computing, I. Lirkov, S. Margenov, and J. Waśniewski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 334–342.

[62] A. J. Nebro, "package org.uma.jmetal.algorithm.multiobjective.abyss," GitHub, Retrieved: December 2020. [Online]. Available: https://github.com/jMetal/jMetal/blob/master/jmetal-algorithm/src/main/java/org/uma/jmetal/algorithm/multiobjective/abyss/ABYSS.java

[63] F. Bourennani, Leadership-Based Multi-Objective Optimization with Applications in Energy Systems. University of Ontario Institute of Technology (Canada), 2013, Retrieved: December 2020. [Online]. Available: https://ir.library.dc-uoit.ca/bitstream/10155/396/1/Bourennani_Farid.pdf

A Large Scale Empirical Evaluation of the Accuracy of Function Points Estimation Methods

Luigi Lavazza
Dipartimento di Scienze teoriche e Applicate
Università degli Studi dell'Insubria
21100 Varese, Italy
Email: luigi.lavazza@uninsubria.it
Geng Liu
School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
Email: liugeng@hdu.edu.cn

***Abstract*—Functional size measures of software—especially Function Points—are widely used, because they provide an objective quantification of software size in the early stages of development, as soon as functional requirements have been analyzed and documented. Unfortunately, in some conditions, performing the standard Function Point Analysis process may be too long and expensive. Moreover, functional measures could be needed before functional requirements have been elicited completely and at the required detail level. To solve this problem, many methods have been invented and are being used to estimate the functional size based on incomplete or not fully detailed requirements. Using these methods involves a trade-off between ease and timeliness on one side and accuracy on the other side. In fact, estimates are always affected by some error; knowing the magnitude of estimation errors that characterize the estimates provided by a given method is of great importance to practitioners. This paper reports the results of an empirical study devoted to evaluate the accuracy of estimates provided by Function Points estimation methods. The results of the study show that some of the evaluated methods—including the Early & Quick Function Points, the ISBSG average and the NESMA estimated method)—provide estimates that are accurate enough for practical usage, while some other methods appear quite inaccurate.**

Keywords*–*Function Points; IFPUG; Function point Analysis; Functional Size Measurement; Functional Size Estimation; NESMA Estimated; NESMA Indicative; Early Size Estimation.

I. INTRODUCTION

This paper extends the results provided in a previous paper [1], in which we started estimating the accuracy methods for estimating functional size measures.

The availability of accurate functional size measures can help software companies plan, monitor, estimate development costs, and control software development processes. So, the availability of accurate functional size measures may provide software companies with competitive advantages over other companies.

Among the functional size measurement methods that have been proposed, Function Point Analysis (FPA) [2] is by far the most popular. The International Function Points User Group (IFPUG) took charge of maintaining FPA and publishes the official Function Point counting manual [3].

In some conditions, performing the standard FPA process may be too long and expensive. Moreover, standard FPA can be applied only after the completion of the software requirements elicitation stage, while functional measures could be needed earlier, i.e., before functional requirements have been elicited completely and at the required detail level.

Therefore, many methods were invented and used to provide *estimates* of functional size measures based on less or coarser grained information than required by standard FPA.

Among the most widely known and used methods are the NESMA methods [4], Early&Quick Function Points (EQFP) [5], [6], the Tichenor ILF Model [7], the simplified FP (sFP) approach [8], the ISBSG distribution model and the ISBSG average weights model, the latter two methods being based on the analysis of the ISBSG dataset [9]. Actually, the NESMA methods were adopted by IFPUG as the preferred methods for early estimation of Function Point size [10] (IFPUG renamed the NESMA Estimated method 'High Level FPA Method' and the NESMA Indicative method as 'Indicative FPA Method'; nonetheless, in this paper we use the original NESMA names).

Inevitably, all the early functional size estimation methods involve some estimation error. Hence, project managers need to know—at least approximately—the magnitude of the potential error that affects size estimates.

Not surprisingly, several researchers and practitioners evaluated the accuracy of the proposed functional size estimation methods (as described in Section VII). However, most evaluations were based on academic software projects or small datasets, hence most evaluations cannot be considered very reliable, and they are hardly generalizable. In order to assess the actual value of the FP estimation methods for industry, it is necessary to perform an experimental evaluation based on a large dataset collecting measures from industrial settings.

In this paper, we present the experimental evaluation of the accuracy of several estimation methods, based on a dataset that includes data from 479 software development projects in the finance and banking domain. The size of the dataset and the real-life nature of the data make the analysis presented here the most exhaustive and reliable published evaluation of Function Point estimation methods.

This paper extends the results provided in a previous paper [1], in which we estimated only the accuracy of NESMA methods. This paper also replicates a previous empirical study [11], in which we evaluated the accuracy of the same estimation methods studied in this paper. In [11] we were able to study only 18 project data, while here we analyze a dataset collecting measures from 479 software projects. The results provided here are relevant in the following respects:

- We were able to confirm that the 'NESMA Estimated' method is among the most accurate proposed methods.
- Some methods appear definitely not accurate; more precisely, their accuracy appears too low, even for usage in the earliest phases of software development.
- Some methods appear even more accurate than the NESMA method, hence project managers can consider using these methods, instead of the NESMA Estimated, even though the latter was chosen as the 'official' estimation methods by IFPUG.

The study described in this paper is of practical relevance because of the following reasons:

- One of the main hindrances perceived by software developers when applying FPA is that it is relatively difficult and expensive, and requires specifically trained and certified measurers. To this end, estimation methods were proposed not only to make FPA applicable before functional requirement specifications are complete and detailed, but also as an easier—hence quicker and cheaper—alternative to FPA, to be applied even when requirements have been fully documented.
- The application of standard functional size measurement does not fit in agile development processes, which are being adopted by software development organizations to a continuously increasing degree. On the contrary, estimation methods are simple enough to fit into agile development processes. Consider for instance that the data required to estimate Function Points using methods like the Early & Quick Function Points or the NESMA methods can be easily derived from the 'stories' that are often used to describe requirements in agile contexts.
- Anticipating the moment when (approximated) functional measures become available means that development effort estimations can be anticipated as well. This is often of great importance, e.g., in bidding processes.

The rest of the paper is organized as follows. Section II briefly describes FPA. Section III gives a concise introduction to the FP estimation measurement methods studied in this paper. Section IV describes the empirical study. Section V proposes some considerations on the outcomes of the empirical study. Section VI discusses the threats to the validity of the study. Section VII illustrates the related work. Finally, Section VIII draws some conclusions and outlines future work.

II. Function Point Analysis

This section provides a concise introduction to FPA. Readers are referred to the official documentation [3] for further details.

Function Point Analysis was originally introduced by Albrecht to measure the size of data-processing systems from the end-user's point of view, with the goal of estimating the development effort [2].

The initial interest sparked by FPA, along with the recognition of the need for maintaining FPA counting practices, led to founding the IFPUG (International Function Points User Group). The IFPUG (http://www.ifpug.org/) maintains the counting practices manual [3], provides guidelines and examples, and oversees the standardization of the measurement method.

The IFPUG method is an ISO standard [12] in its "unadjusted" version. The adjustment factor originally proposed by Albrecht and endorsed by IFPUG is meant to obtain measures more apt for effort estimation, by accounting for factors not dealing with functional requirements, namely with product and process features that do not belong to the notion of functional size. As such, the adjustment was not accepted by ISO; so, throughout the paper we refer exclusively to unadjusted FP (UFP), even when we omit adjective "unadjusted."

Albrecht's basic idea—which is still at the basis of the IFPUG method—is that the "amount of functionality" released to the user can be evaluated by taking into account 1) the data used by the application to provide the required functions, and 2) the transactions (i.e., operations that involve data crossing the boundaries of the application) through which the functionality is delivered to the user. Both data and transactions are evaluated at the conceptual level, i.e., they represent data and operations that are relevant to the user. Therefore, IFPUG Function Points are counted on the basis of the user requirements specification. The boundary indicates the border between the application being measured and the external applications and user domain.

FURs are modeled as a set of base functional components (BFCs), which are the measurable elements of FURs: each of the identified BFCs is measured, and the size of the whole application is obtained as the sum of the sizes of BFCs.

The IFPUG model of a software application to be measured is shown in Figure 1. IFPUG BFCs are data functions (also known as logical files), which are classified into internal logical files (ILF) and external interface files (EIF), and elementary processes (EP)—also known as transaction functions—which are classified into external inputs (EI), external outputs (EO), and external inquiries (EQ), according to the activities carried out within the process and its main intent.

So, the functional size of a given application, expressed in unadjusted Function Points, $Size_{UFP}$, is given by the sum

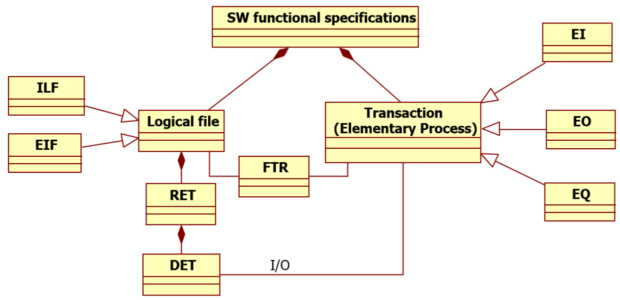


Figure 1. The IFPUG model of software.

of the sizes of the different types of functions, as shown in equation (1).

$$Size_{UFP} = \sum_{f \in ILFs} Size_{ILF}(f) + \sum_{f \in EIFs} Size_{EIF}(f) + \sum_{f \in EIs} Size_{EI}(f) + \sum_{f \in EOs} Size_{EO}(f) + \sum_{f \in EQs} Size_{EQ}(f) \quad (1)$$

In equation (1), $ILFs$ is the set of all data functions of type ILF, EIs is the set of all transactional functions of type EI, etc. Also, $Size_X(f)$ is the weight of function f, which depends on its complexity, and its type ($X \in \{ILF, EIF, EI, EO, EQ\}$, as described in Table I.

TABLE I. FPA WEIGHT TABLE.

Function type	Low	Complexity Average	High
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

The complexity of a data function (ILF or EIF) depends on the RETs (Record Element Types), which indicate how many types of information (e.g., sub-classes, in object-oriented terms) can be contained in the given logical data file, and DETs (Data Element Types), which indicate how many types of elementary information (e.g., attributes, in object-oriented terms) can be contained in the given logical data file.

The complexity of a transaction depends on the number of file types references (FTRs)—i.e., the number of types of logical data files used while performing the required operation—and the number of DETs—i.e., the number of types of elementary data—that the considered transaction sends and receives across the boundaries of the application.

Details concerning the determination of complexity can be found in the official documentation [3].

The core of FPA involves three main activities:

1) Identifying data and transaction functions.
2) Classifying data functions as ILF or EIF, and transactions as EI, EO or EQ.
3) Determining the complexity of each data or transaction function.

The first two of these activities can be carried out even if the FUR have not yet been fully detailed. On the contrary, the last activity requires that all details are available, so that such details can be analyzed, to get the number of RET or FTR and DET involved in every function. In the next section, several FP estimation methods are presented. All of these method do not require activity 3 above, thus allowing for size estimation when FUR are not fully detailed. In addition, since activity 3 is the most effort- and time-consuming, the presented estimation methods are relatively fast and cheap, with respect to full-fledged FPA.

III. FUNCTION POINTS ESTIMATION METHODS

In this section we briefly present the FP estimation methods mentioned in the introduction.

A. The NESMA methods

The NESMA methods were proposed to get an estimate of the functional size of a given application without analyzing data and transactions in detail [4].

There are two NESMA estimation methods: the NESMA Indicative method and the NESMA Estimated method.

The former estimates size (*EstSize*) based on the number of ILF (*#ILF*) and the number of EIF (*#EIF*), as follows:

$$EstSize = \#ILF \times W_{ILF} + \#EIF \times W_{EIF}$$

where W_{ILF} is 25 or 35 and W_{EIF} is 10 or 15, depending on ILF and EIF being identified based on a data model in third normal form or not, respectively.

The process of applying the NESMA indicative method involves only identifying logic data and classifying them as ILF or EIF. Accordingly, it requires less time and effort than standard FPA. However, the NESMA Indicative method is quite rough in its computation: the official NESMA counting manual specifies that errors in functional size with this approach can be up to 50%.

The NESMA Estimated method requires the identification and classification of all data and transaction functions, but does not require the assessment of the complexity of each function: Data Functions (ILF and EIF) are assumed to be of low complexity, transactions (EI, EQ and EO) are assumed to be of average complexity. Hence, estimated size is computed as follows:

$$EstSize = 7\ \#ILF + 5\ \#EIF + 4\ \#EI + 5\ \#EO + 4\ \#EQ$$

IFPUG adopted the NESMA methods as the official early function point analysis methods [10].

B. The Early & Quick Function Points method

One of the most well-known approach for simplifying the process of FP counting is the Early & Quick Function Points (EQFP) method [5], [6]. EQFP descends from the consideration that estimates are sometimes needed before requirements analysis is completed, when the information on the software to be measured is incomplete or not sufficiently detailed. Since several details for performing a correct measurement following the rules of the FP manual [3] are not used in EQFP, the result is a less accurate measure. The trade-off between reduced measurement time and costs is also a reason for adopting the EQFP method even when full specifications are available, but there is the need for completing the measurement in a short time, or at a lower cost. An advantage of the method is that different parts of the system can be measured at different detail levels: for instance, a part of the system can be measured following the IFPUG manual rules [3], [12], while other parts can be measured on the basis of coarser-grained information, possibly considering analogy with previously measured software parts. The EQFP method is based on the classification of the processes and data of an application according to a hierarchy (see Figure 2 [6]).

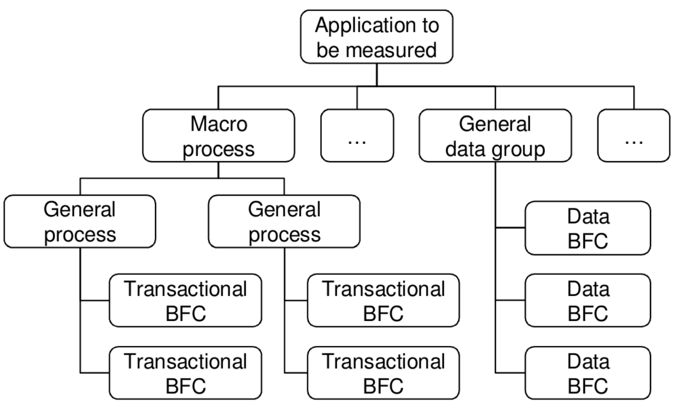


Figure 2. Functional hierarchy in the Early & Quick FP technique.

Transactional BFC and Data BFC correspond to IFPUG's elementary processes and LogicData, while the other elements are aggregations of processes or data groups. The idea is that if you have enough information at the most detailed level you count FP according to IFPUG rules; otherwise, you can estimate the size of larger elements (e.g., General or Macro processes) either on the basis of analogy (e.g., a given General process is "similar" to a known one) or according to the structured aggregation (e.g., a General process is composed of 3 Transactional BFC). By considering elements that are coarser-grained than the FPA BFC, the EQFP measurement process leads to an approximate measure of size in IFPUG FP.

Tables taking into account the previous experiences with the usage of EQFP are provided to facilitate the task of assigning a minimum, maximum and most likely quantitative size to each component. For instance, Table II provides minimum, maximum and most likely weight values for generic (i.e., not weighted) functions as given in [6]. The time and effort required by the weighting phases are thus saved. Such saving can be relevant, since weighting a data or transaction function requires analyzing it in detail.

TABLE II. EQFP: FUNCTION TYPE WEIGHTS FOR GENERIC FUNCTIONS.

	Weight		
Function type	Low	Likely	High
Generic ILF	7.4	7.7	8.1
Generic EIF	5.2	5.4	5.7
Generic EI	4	4.2	4.4
Generic EO	4.9	5.2	5.4
Generic EQ	3.7	3.9	4.1

C. Tichenor method

The Tichenor ILF Model [7] bases the estimation of the size on the number of ILF via the following formula for transactional system (for batch systems, Tichenor proposes a smaller multiplier): $EstSize = 14.93\ \#ILF$

This model assumes a distribution of BFC with respect to ILF as follows: EI/ILF = 0.33, EO/ILF = 0.39, EQ/ILF = 0.01, EIF/ILF = 0.1. If the considered application features a different distribution, the estimation can be inaccurate.

The fact that a method based only on ILF requires a given distribution for the other BFC is not surprising. In fact, the size of the application depends on how many transactions are needed to elaborate those data, and the number of transaction cannot be guessed only on the basis of the number of ILF, as it depend on the number of ILF just very loosely. Instead of allowing the user to specify the number of transactions that are needed, the Tichenor method practically imposes that the number of transactions complies with the distribution given above.

D. Simplified FP

The simplified FP (sFP) approach assumes that all BFC are of average complexity [8], thus:

$$EstSize = 4\ \#EI + 5\ \#EO + 4\ \#EQ + 10\ \#ILF + 7\ \#EIF$$

E. ISBSG distribution model

The analysis of the ISBSG dataset yielded the following distribution of BFC contributions to the size in FP: ILF 22.3%, EIF 3.8%, EI 37.2%, EO 23.5%, EQ 13.2%

The analysis of the ISBSG dataset also shows that the average size of ILF is 7.4 UFP. It is thus possible to compute the estimated size on the basis of the number of ILF as follows:

$$EstSize = 7.4\ \#ILF \frac{100}{22.3}$$

The same considerations reported above for the Tichenor model apply. If the application to be measured does not fit the distribution assumed by the ISBSG distribution model, it is likely that the estimation will be inaccurate.

F. ISBSG average weights

This model is based on the average weights for each BFC, as resulting from the analysis of the ISBSG dataset [9], which contains data from a few thousand projects. Accordingly, the ISBSG average weights model suggests that that the average function complexity is used for each BFC, thus

$$EstSize = 4.3\ \#EI + 5.4\ \#EO + 3.8\ \#EQ + 7.4\ \#ILF + 5.5\ \#EIF$$

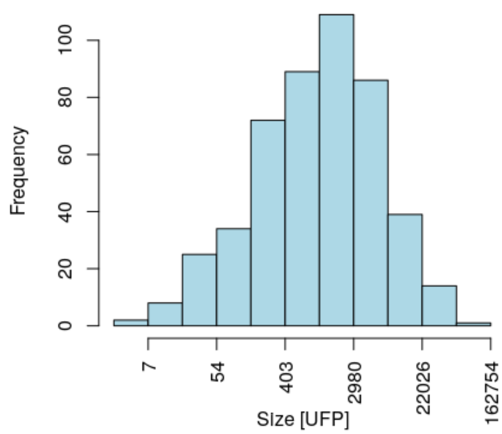


Figure 3. Distribution of projects' sizes in the studied dataset (note the logarithmic scale on the x axis).

IV. The Empirical Study

In this section, we first provide a brief description of the dataset used for the evaluation of functional size estimation methods; then, we describe the analysis and the resulting accuracy evaluations.

A. The Dataset

We use a dataset that collects data from 479 software applications developed and used by a Chinese financial enterprise.

Descriptive statistics of the dataset are given in Table III.

TABLE III. Descriptive statistics of dataset.

	Standard UFP
Mean	3554
St. Dev.	6673
Median	1155
Min	4
Max	80880

The distribution of size is given in Figure 3. Specifically, it is worth noting that the studied dataset contains relatively few data from small projects (32 projects not greater than 50 UFP) and relatively few data concerning very large projects (17 projects bigger than 20000 UFP).

B. The Analysis

For each of the 479 project in the dataset, we computed the estimated size, using each of the estimation method described in Section III.

To assess the estimates, in Figures 4 and 5 we plot the values of the estimates with respect to the actual size measured according to the standard IFPUG counting manual [3]. Specifically, Figure 4 illustrates the estimates yielded by the most accurate methods, while Figure 5 shows the estimates yielded by the less accurate methods.

In Figures 4 and 5, we also draw the estimated size = actual size line: if the estimates were perfect, all the points would lie on the line.

Looking at Figure 4, it is easy to see that the estimates provided by the NESMA Estimated, EQFP, Simplified FP, and ISBSG average methods are close to the x=y line, thus indicating fairly accurate estimates. On the contrary, in Figure 5 it can be observed that the estimates provided by the NESMA indicative (normalized and not normalized), Tichenor, and ISBSG distribution methods are widely spread, with many points definitely far from the x=y line, thus indicating that many estimates are quite inaccurate.

To better appreciate the accuracy of estimates, in the Appendix, Figures 13 and 14 illustrate estimates for projects having size in the [50, 20000] UFP range.

Figures 5 and 14 seem to indicate that the NESMA indicative (normalized and not normalized), Tichenor, and ISBSG distribution methods are rather inaccurate. To better investigate this indication, in Figure 6 we give the boxplots that illustrate the distributions of errors (errors are defined as the difference actual size minus estimated size). To keep the figure readable, outliers (i.e., the biggest errors) are not shown.

It is easy to see that NESMA Estimated, EQFP, Simplified FP, and ISBSG average weight methods have narrow errors distributions, close to the zero-error line. On the contrary, the other methods feature widespread distributions, in some cases centered on positive errors. Specifically, the Tichenor method provides the worst accuracy and tends to underestimate (for over 75% of the projects, estimates are smaller than the actual size). Similar considerations—although to a lesser extent—apply to the NESMA indicative (normalized) and ISBSG distribution methods. The NESMA indicative (not normalized) method does not tend to underestimate nor overestimate, but its estimation errors are often quite large.

To better study the errors of the best methods, in Figure 7 we give the boxplots of the estimation errors of the NESMA Estimated, EQFP, Simplified FP, and ISBSG average weight methods.

Figure 7 shows that the NESMA Estimated method tends to underestimates, while the Simplified FP method tends to overestimate. On the contrary, the EQFP and the ISBSG average weights methods appear symmetrically distributed with respect to the zero-error line.

It is worth noticing that size underestimation is quite dangerous, since it may lead to underestimating development cost

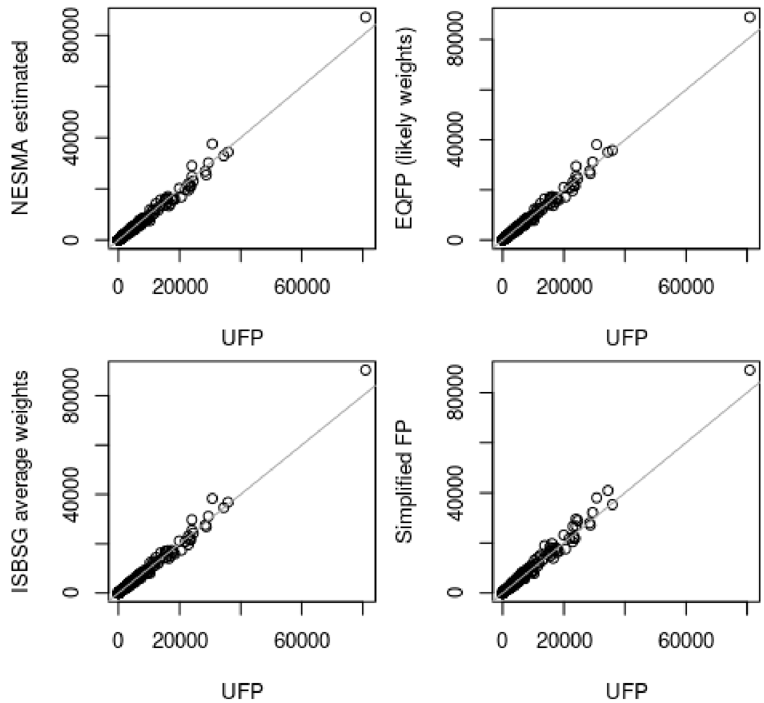


Figure 4. Standard IFPUG UFP measures vs. estimates provided by most accurate methods.

and duration, hence to defining not realistic plans, that usually lead to bad project organization, understaffing, overspending, etc.. Accordingly, using the NESMA Estimated method could be risky.

This observation concerning the NESMA Estimated method is particularly interesting because the NESMA Estimated method is also the one that most frequently provides the best accuracy in the lot of estimation methods, as shown in Figure 8 (the most accurate estimate here is the one characterized by the least absolute error). So, practitioners must consider that the NESMA Estimated method is likely to provide the smallest estimation errors, but is also most frequently underestimating.

C. Accuracy Evaluation

It is now necessary to evaluate quantitatively the accuracy of estimates. First of all—as suggested by Shepperd and MacDonell [13]—we checked whether estimates perform better than "baseline" models. Shepperd and MacDonell [13] proposed that the accuracy of a given estimation method be measured via the Mean Absolute Residual (MAR): $MAR = \frac{1}{n}\sum_{i=1..n}|y_i - \hat{y}_i|$, where y_i is the actual value of the i^{th} observation, and $\hat{y}_i$ is its estimate. Shepperd and MacDonell suggest to use random estimation, based solely on the known (actual) values of previously measured applications, as a baseline model. Shepperd and MacDonell observed also that the value of the 5% quantile of the random estimate MARs can be interpreted like α for conventional statistical inference, that is, any accuracy value that is better than this threshold has a less than one in twenty chance of being a random occurrence. Accordingly, the MAR of a proposed model should be compared with the 5% quantile of the random estimate MARs, to make us reasonably sure that the model is actually more accurate than random estimation.

Lavazza and Morasca [14] proposed to use a "constant model," where the estimate of the size of an application is given by the mean size of the other applications.

With our dataset, the MAR of the constant model is 3864 UFP, while the 5% quantile of absolute residuals for random estimates is 4566 UFP. The MARs of estimates are given in Table IV, together with MdAR (the median of absolute errors) and MMRE (the mean of relative absolute errors).

TABLE IV. ACCURACY INDICATORS FOR ESTIMATION METHODS.

Method	MAR	MdAR	MMRE
NESMA Estimated	315	73	0.102
NESMA Indic. (not norm.)	2135	435	0.629
NESMA Indic. (norm.)	1931	425	0.523
EQFP (likely weights)	299	79	0.103
Tichenor	2363	638	0.648
ISBSG distribution est.	2045	451	0.589
ISBSG average weights	298	82	0.103
Simplified FP	411	112	0.141

According to Table IV, all the evaluated methods satisfy the necessary conditions for being considered acceptable estimation methods.

Concerning the accuracy of estimates, in Figure 9 the distribution of absolute errors (excluding the biggest ones,

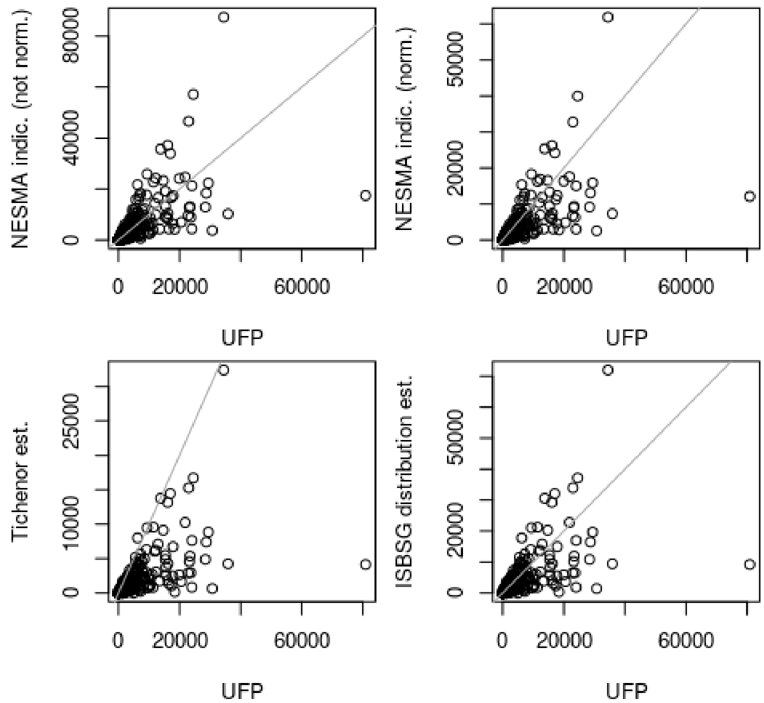


Figure 5. Standard IFPUG UFP measures vs. estimates provided by less accurate methods.

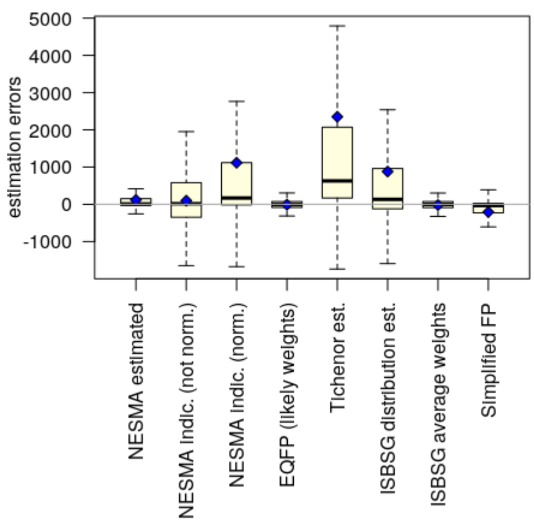


Figure 6. Boxplots of estimation errors (no outliers shown).

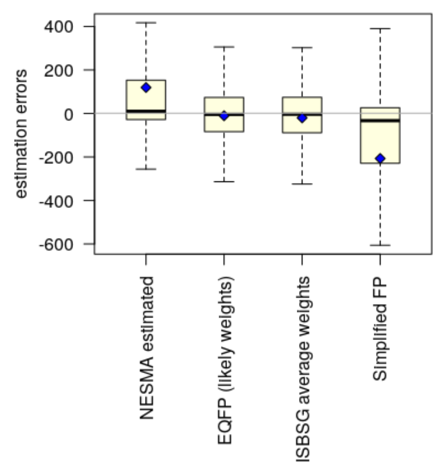


Figure 7. Boxplots of selected methods' estimation errors (no outliers shown).

to keep the figure readable) is given. The blue diamond is the mean, i.e., the MAR of the estimates. Figure 9 confirms the observation already reported in Section IV-B, i.e., even not considering the biggest errors, it appears that NESMA Indicative, Tichenor and ISBSG distribution methods feature many fairly inaccurate estimates.

As a result of the previously described findings, it seems that NESMA Estimated, EQFP, ISBSG average weights and Simplified FP methods qualify as the best candidates for size estimation, therefore these methods deserve most attention. Specifically, we should evaluate whether these methods are acceptable for functional size estimation. To this end, in Figure 10 the boxplot of the absolute errors of the estimates obtained via these methods are illustrated. Note that in Figure 10—unlike in Figure 9—all the absolute errors are

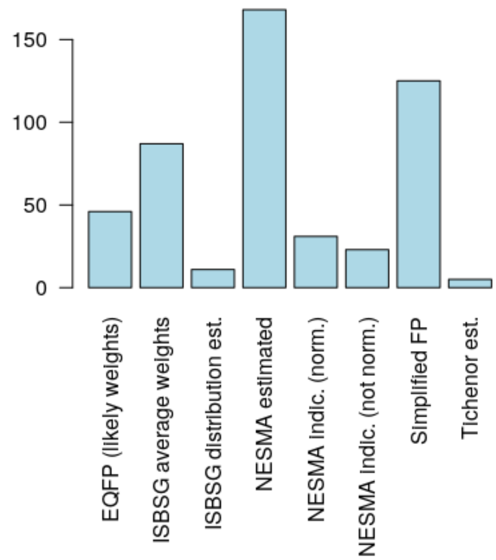


Figure 8. How many times every method yielded the best estimate.

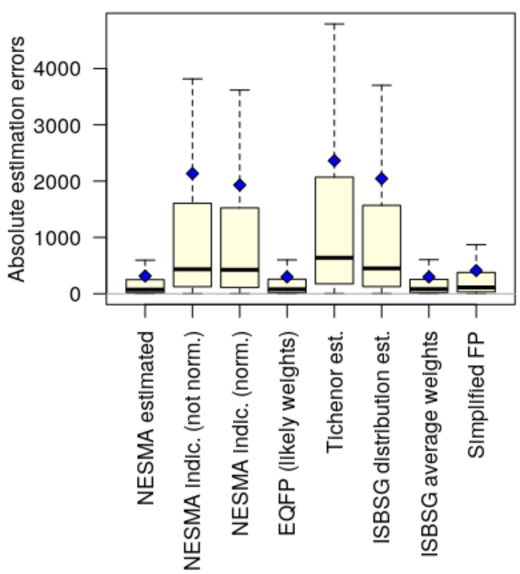


Figure 9. Boxplots of absolute errors (outliers not shown).

represented.

Figure 10 shows quite clearly that while the considered methods feature a fairly small absolute error, all methods are characterized by some quite big errors. To evaluate whether these errors are acceptable, we can consider that in general, relatively large estimation errors are deemed acceptable in very large projects.

To help practitioners appreciate the "importance" of errors with respect to the size of the estimated project, in Figure 11 we give the boxplots representing the distributions of relative errors (the relative error of an estimate is the estimation error divided by the actual size). To keep Figure 11 readable, the biggest errors are not shown.

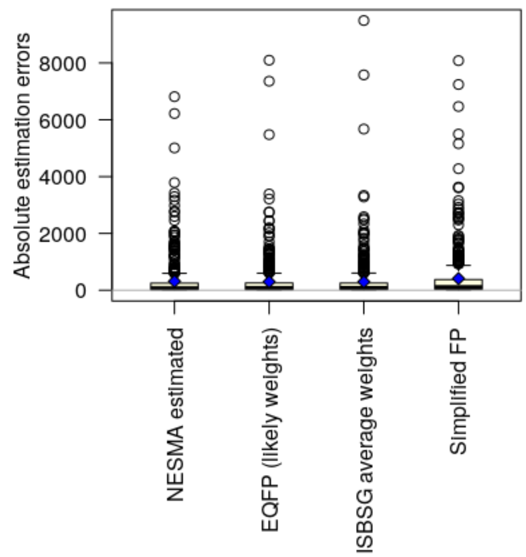


Figure 10. Boxplots of all absolute errors for selected methods.

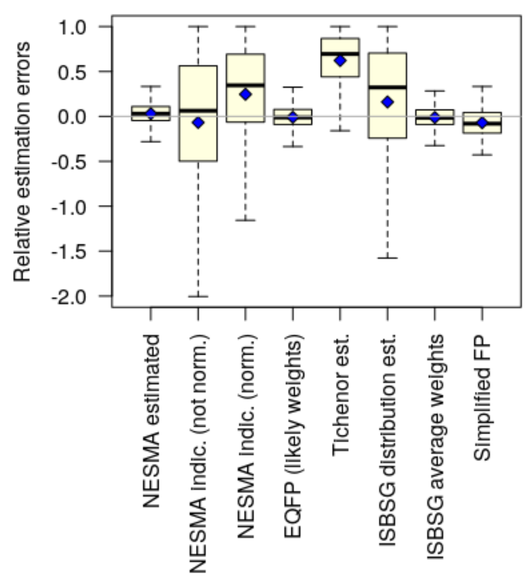


Figure 11. Boxplots of relative errors (no outliers).

Figure 11 shows that the estimation errors caused by the NESMA Indicative, Tichenor and ISBSG distribution methods are quite large also in relative terms. Specifically, these methods feature several relatively large underestimations: all the three mentioned methods underestimate no less that 25% of the projects by 50% or more. Even though the acceptability of estimates is largely subjective, we have strong doubts that any practitioner would find these levels of accuracy acceptable.

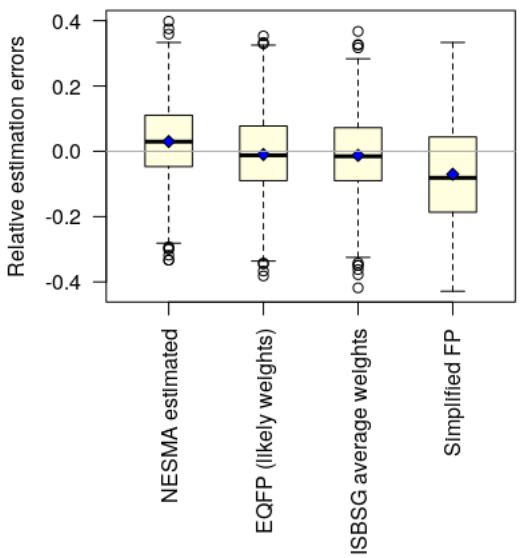


Figure 12. Boxplots of all relative errors for selected methods.

Considering the other methods, Figure 12 shows the boxplots of all the relative estimation errors. Figure 12 indicates that all the selected methods perform well. For all methods, the great majority of errors is in the [-20%, +20%] range, which is generally considered acceptable, especially when the estimate is performed in the earliest phases of the software lifecycle, when user requirements have not yet been fully analyzed.

V. Considerations on the Outcomes of the Empirical Study

The results obtained via the empirical study support a few considerations, which we illustrate in the following sections.

A. Methods based on the knowledge of data functions

The NESMA Indicative, Tichenor and ISBSG distribution estimation methods are all based on the knowledge of the number of data functions only. The NESMA Indicative method also requires that logical data files are classified as ILF or EIF, which is usually quite easy, since one has just to check whether the application being measured maintains (i.e., creates, modifies or deletes) the considered data files or not. On the contrary, the other methods considered in this paper base their estimates on the knowledge or transactional functions as well.

Since the NESMA Indicative, Tichenor and ISBSG distribution estimation methods use the knowledge of the number of data functions only, it is not surprising that they provide a level of accuracy that is definitely lower with respect to methods that account also for transactions.

So, the question is now whether it is reasonable to use estimates that are based exclusively on the knowledge of logical data files. First of all, we can observe that the question makes sense, since in many cases requirements analysis starts with domain analysis, which provides reliable indications concerning the data the software application will have to deal with. In such conditions, no method appears to yield acceptably accurate estimates.

B. Methods based on the knowledge of data and transaction functions

In order to estimate size using the NESMA Estimated, EQFP, ISBSG average weights or Simplified FP methods, you have to know both the number of data functions (and how they are classified into ILF and EIF) and the number of transactions (and how they are classified in EI, EO and EQ). If you have this knowledge of functional requirements, you can get size estimates whose error is in the [-20%, +20%] range, except in a minority of cases.

Given this, our results show that the Early & Quick FP and the ISBSG average methods are probably preferable, since they avoid both underestimation (which affects the NESMA Estimated method) and overestimation (which affects the Simplified FP method). In addition, the Early & Quick FP and the ISBSG average methods a slightly smaller MAR than the NESMA estimated method (see Table IV).

Actually, both the NESMA Estimated method and the Simplified FP method are based on assumptions that have never been verified. The former method assumes that data functions are all of low complexity and all transactions are of average complexity; the latter method assumes that all functions are of average complexity. Our study shows that none of these assumptions appear totally correct: in fact, the NESMA Estimated method tends to underestimate and the Simplified FP method tends to overestimate (note that, by definition, for a given application, the Simplified FP estimate will always be greater than the NESMA Estimated estimate).

The remaining methods, i.e., the EQFP and the ISBSG average weights methods are based on statistical analysis of large datasets. They propose for each function type the average weight observed. As a consequence, the definition of the two methods are very similar, and the estimates they provide are usually very close. This makes quite hard to choose one of the two methods. To this end, an interesting observation is that the EQFP method provides not only 'likely' weights for function types, but also low and high values (see Table II). By sizing all functions with the low weights and with the high weights gives a sort of confidence interval that could cope with the inherent uncertainty of the estimation due to the lack of details concerning functional requirements.

As a final consideration, the fact that both the EQFP and the ISBSG average weights methods performs well shows that organizations that have large enough collections of historical data could build their own estimation method following the same process as the EQFP and the ISBSG average weights methods: compute the average weight of each function type from the historical dataset and use such weights to estimate the size of new developments.

VI. Threats to Validity

Given the type of study we presented, there are two main threats to validity that need attention.

First, we should consider the correctness of the given data. In fact, the data in the analyzed dataset were derived from the analysis and measurement of functional requirements: both analysis and measurement could be affected by error, which would end up in the dataset. Concerning this threat, we are reasonably sure that the used data are of good quality, since they were collected by professionals in industrial contexts where functional size measures are quite important, hence great attention is posed in the measurement activities.

Second, we need to consider external validity, i.e., whether we can generalize the results of our study outside the scope and context that characterize the considered software projects. On the one hand, our dataset is much larger than the datasets usually involved in software engineering empirical studies; besides, our dataset includes data from fairly large projects (e.g., over 20000 FP). In this sense, our dataset represents a large and varied enough sample. On the other hand, all the considered projects are from the economic, financial and banking domain, hence we cannot be sure that the results of our study apply equally well in other domains. In this respect, readers are reminded that previous studies show some difference in accuracy when estimates concern other types of software applications, e.g., real-time applications [11].

VII. Related work

Meli and Santillo were among the first to recognize the need for comparing the various functional size methods proposed in the literature [15]. To this end, they also provided a benchmarking model.

Popović and Bojić compared different functional size measures—including NESMA Indicative and Estimated—by evaluating their accuracy in effort estimation in various phases of the development lifecycle [16]. Not surprisingly, they found that the NESMA Indicative method provided the best accuracy at the beginning of the project. With respect to Popović and Bojić, we made two quite different choices: the accuracy of the method is evaluated against the actual size of the software product and—consistently—all the information needed to perform measurement is available to all processes.

Santillo suggested probabilistic approaches, where the measurer can indicate the minimum, medium and maximum weight of each BFC, together with the expected probability that the weight is actually minimum, medium or maximum [17]. This leads to estimate not only the size, but also the probability that the actual size is equal to the estimate.

NESMA defined the application of FPA in the early phases of the application life cycle, and recognizes three function point analysis methods: Detailed function point analysis (currently corresponding to IFPUG measurement), Estimated function point analysis, and Indicative function point analysis. Using a database of over 100 developed and implemented applications, NESMA empirically evaluated the accuracy of the Estimated and Indicative FPA approximation methods [18]. The results showed that size measures of the high-level function point analysis and the detailed function point analysis are very close. Moreover, Indicative function point analysis gives a surprisingly good estimate of the size of several applications.

van Heeringen described the size accuracy—as well as the difference in measurement effort—of the NESMA Estimated and NESMA Indicative methods, by measuring 42 projects [4]. The results show that the estimation error of NESMA Estimated was in the [-6%, +15%] range, with average 1.5%; the estimation error of NESMA Indicative was in the [-15%, +50%] range with average 16.3%. In both cases the estimation error was evaluated with respect to detailed measurement.

Wilkie et al. [19] used five commercial projects to evaluate the cost-benefit trade-off of size measurement with respect to size estimation; they concluded that whilst the NESMA Indicative method was insufficiently accurate for the involved commercial organization, the NESMA Estimated approach was definitely viable.

IFPUG adopted NESMA methods for early "high-level" size estimation [10]. IFPUG suggested that 1) The High Level FPA method can be used to size an application early in the software development life cycle; 2) The High Level FPA method can also be applied as an alternative to standard FPA estimate (the outcome is not significantly different, while the estimation time is considerably shorter); 3) The indicative FPA method may be used to get a very fast, rough indication of the size of an application, but it is not suited for contractual commitments.

Lavazza et Liu [11] used 7 real-time applications and 6 non real-time applications to evaluate the accuracy of the EQFP [5] and NESMA methods with respect to full-fledged Function Point Analysis. The results showed that the NESMA Indicative method yields the greatest errors. On the contrary, the NESMA Estimated method yields size estimates that are close to the actual size. The NESMA Indicative method is generally outperformed by the other methods. The NESMA Estimated method proved fairly good in estimating both Real-Time and non Real-Time applications.

Morrow et al. used a dataset of 11 projects to evaluate the quality of sizing estimates provided by NESMA methods [20]. They also adapted NESMA methods' general principles to enhance their accuracy and extent of relevance, and empirically validated such an adapted approach using commercial software projects.

The main limitations of the mentioned research are that most studies used small datasets containing data concerning little projects of not industrial nature. In our paper, we evaluate measurement accuracy of the NESMA method with respect to FPA method over a dataset containing data from 479 industrial projects, of which several are above 10000 FP.

Ochodek proposed a method to approximate IFPUG FPA functional size in an automatic way, based on given UML use-case diagrams or a list of use-case names [21].

Meli proposed Simple Function Points (SiFP), as an alternative to standard IFPUG FP [22]. SiFP have been evaluated via empirical studies [23], [24] that showed that they appear suitable replacements of standard FP.

VIII. Conclusions

In this paper we addressed the evaluation of the accuracy of functional size estimates that can be achieved via several Function Points estimation methods. To this end, we compared functional size measures obtained via the standard IFPUG Function Point Analysis process with estimates obtained via the considered estimated methods. Both measures and estimates were computed for a dataset containing data from 479 software projects. Based on the results of the analysis, we can draw a few relevant conclusions:

- The methods that use knowledge of both data and transaction functions provide estimates that are much more accurate than those provided by methods that use only knowledge of data functions.
- The methods that use knowledge of both data and transaction functions, namely the NESMA Estimated, Early & Quick FP, ISBSG average weights and the Simplified FP methods, provide estimates that are mostly in the [-20%, -20%] error range, hence their estimates are likely accurate enough in most cases.
- The NESMA Estimated method tends to underestimate. This can be dangerous, since at the initial stages of development one could be induced to believe that the development process will be shorter and cheaper than actually required.
- The Simplified FP method tends to overestimate.
- With the Early & Quick FP and the ISBSG average weights methods, the probabilities of getting underestimates and overestimates appear approximately equal.

Future work includes experimenting with new estimation methods that can be derived from the available dataset, and investigating whether and how estimation accuracy can be improved.

Acknowledgment

Parts of this work have been supported by the “Fondo di ricerca d’Ateneo” funded by the Università degli Studi dell’Insubria, by Zhejiang Provincial Science Foundation of China under grant no. LY19F020046, and by the Chinese Scholarship Council under grant no. 201708330399.

References

[1] L. Lavazza and G. Liu, “An Empirical Evaluation of the Accuracy of NESMA Function Points Estimates,” in International Conference on Software Engineering Advances (ICSEA 2019), 2019.

[2] A. J. Albrecht, “Measuring application development productivity,” in Proceedings of the joint SHARE/GUIDE/IBM application development symposium, vol. 10, 1979, pp. 83–92.

[3] International Function Point Users Group (IFPUG), “Function point counting practices manual, release 4.3.1,” 2010.

[4] H. van Heeringen, E. van Gorp, and T. Prins, “Functional size measurement-accuracy versus costs-is it really worth it?” in Software Measurement European Forum (SMEF 2009), 2009.

[5] L. Santillo, M. Conte, and R. Meli, “Early & Quick function point: sizing more with less,” in 11th IEEE International Software Metrics Symposium (METRICS’05). IEEE, 2005, pp. 41–41.

[6] DPO, “Early & Quick Function Points for IFPUG methods v.3.1 Reference Manual 1.1,” DPO srl, 2012.

[7] C. Tichenor, “The IRS Development and Application of the Internal Logical File Model to Estimate Function Point Counts,” in IFPUG Fall Conference of Use (ESCOM-ENCRESS 1998). IFPUG, 1998.

[8] L. Bernstein and C. M. Yuhas, Trustworthy Systems Through Quantitative Software Engineering. John Wiley & Sons, 2005.

[9] International Software Benchmarking Standards Group, “Worldwide Software Development: The Benchmark, release 11,” ISBSG, 2009.

[10] A. Timp, “uTip – Early Function Point Analysis and Consistent Cost Estimating,” 2015, uTip # 03 (version # 1.0 2015/07/01).

[11] L. Lavazza and G. Liu, “An empirical evaluation of simplified function point measurement processes,” International Journal on Advances in Software, vol. 6, no. 1& 2, 2013.

[12] International Standardization Organization (ISO), “ISO/IEC 20926: 2003, Software engineering IFPUG 4.1 Unadjusted functional size measurement method Counting Practices Manual,” 2003.

[13] M. Shepperd and S. MacDonell, “Evaluating prediction systems in software project estimation,” Information and Software Technology, vol. 54, no. 8, 2012, pp. 820–827.

[14] L. Lavazza and S. Morasca, “On the evaluation of effort estimation models,” in Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. ACM, 2017, pp. 41–50.

[15] R. Meli and L. Santillo, “Function point estimation methods: a comparative overview,” in Software Measurement European Forum (FESMA 1999), 1999.

[16] J. Popović and D. Bojić, “A comparative evaluation of effort estimation methods in the software life cycle,” Computer Science and Information Systems, vol. 9, 2012.

[17] L. Santillo, “Easy Function Points – ‘Smart’ Approximation Technique for the IFPUG and COSMIC Methods,” in Joint Conf. of the 22nd Int. Workshop on Software Measurement and the 7th Int. Conf. on Software Process and Product Measurement, 2012.

[18] nesma, “Early Function Point Analysis.” [Online]. Available: https://nesma.org/themes/sizing/function-point-analysis/early-function-point-counting/ accessed on November 8, 2020

[19] F. G. Wilkie, I. R. McChesney, P. Morrow, C. Tuxworth, and N. Lester, “The value of software sizing,” Information and Software Technology, vol. 53, no. 11, 2011, pp. 1236–1249.

[20] P. Morrow, F. G. Wilkie, and I. McChesney, “Function point analysis using nesma: simplifying the sizing without simplifying the size,” Software Quality Journal, vol. 22, no. 4, 2014, pp. 611–660.

[21] M. Ochodek, “Functional size approximation based on use-case names,” Information and Software Technology, vol. 80, 2016, pp. 73–88.

[22] R. Meli, “Simple Function Point: a new functional size measurement method fully compliant with ifpug 4. x,” in Software Measurement European Forum, 2011.

[23] L. Lavazza and R. Meli, “An evaluation of Simple Function Point as a replacement of IFPUG function point,” in 2014 joint conference of the international workshop on software measurement and the international conference on software process and product measurement. IEEE, 2014, pp. 196–206.

[24] F. Ferrucci, C. Gravino, and L. Lavazza, “Simple function points for effort estimation: a further assessment,” in Proceedings of the 31st Annual ACM Symposium on Applied Computing, 2016, pp. 1428–1433.

APPENDIX

Figures 13 and 14 illustrate estimates vs. actual values, concerning projects in the [50, 20000] UFP range.

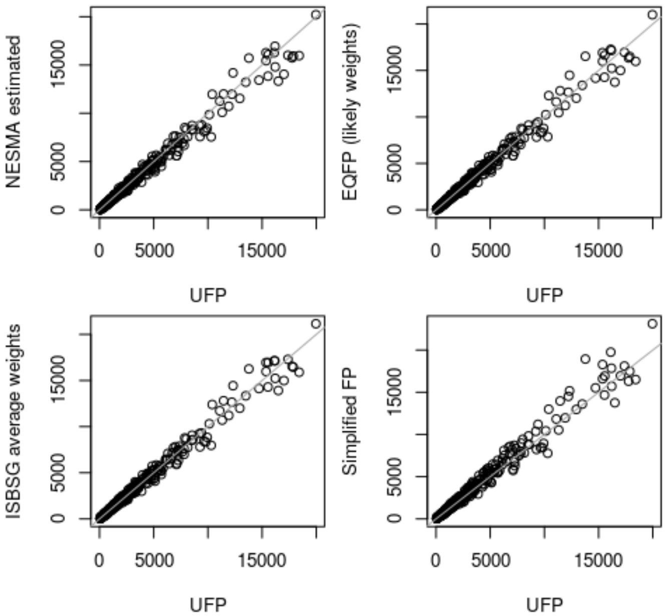


Figure 13. Standard IFPUG UFP measures vs. estimates (no smallest and biggest projects).

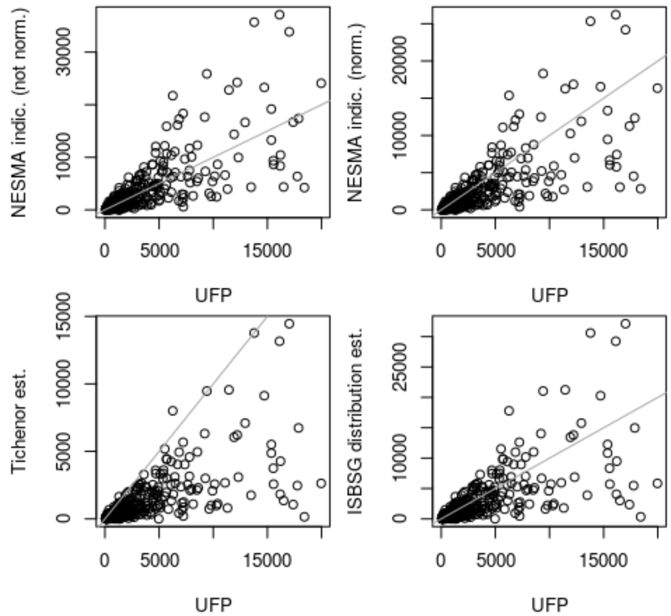


Figure 14. Standard IFPUG UFP measures vs. estimates (no smallest and biggest projects).

Semaphore Implementations for Testing Concurrent Systems using TTCN-3

Bernard Stepien, Liam Peyton
School of Engineering and Computer Science
University of Ottawa, Ottawa, Canada
Email: {bstepien | lpeyton}@uottawa.ca;

Jacob Wieland, Dirk Tepelmann, Dirk Borowski
Spirent Communications
Berlin, Germany
Email: {Jacob.wieland | dirk.tepelmann | dirk.borowski}
@spirent.com

Abstract—**Testing concurrent systems is a complex task. In traditional software unit testing, a test sequence is always composed of a stimulus and its corresponding fully predictable response. With concurrent systems, this simple model no longer holds as the state of the system under test (SUT) changes while several users place their requests. Race conditions are a particularly challenging problem for testing, since they will occur and must be identified, but are very disruptive to the test environment. An easy solution to this problem is to use semaphores that avoid race conditions. Since semaphores do not exist in TTCN-3, we have explored solutions using the TTCN-3 concept of external functions. This allows us to define behavior in the run-time language used by the TTCN-3 compiler, in our case Java, without having to modify the TTCN-3 standard. However, Java semaphores can block other parallel processes which may actually defeat parallelism. Thus, we have explored other solutions based on resource blocking rather than process blocking. This allows two or more concurrent processes to perform operations on different resources and thus achieve more sophisticated concurrency testing.**

Keywords- software testing-concurrent systems; TTCN-3; test oracles; race conditions; semaphores.

I. INTRODUCTION

This paper is a significant extension and update of our previously published SOFTENG 2020 paper that presented preliminary results for advanced techniques for testing concurrent systems [1].

Testing concurrent systems is complex. In traditional software unit testing, a test sequence is always composed of a stimulus and its corresponding fully predictable response [2]. With concurrent systems, this simple model no longer holds as the state of the system under test (SUT) changes while several users place their requests. Race conditions are a particularly challenging problem for testing, since they will occur and must be identified, but are very disruptive to the test environment.

Some definitions and implementations of parallel testing can be found in [3][4][5][6][7]. Obviously there are different kinds of parallel testing. In the previous references, the main concern is to run sequential tests in parallel in order to save time. Instead, we focus on concurrent testing of states in a (SUT) as the test purpose. There are two main categories of concurrent testing:

- Response time testing when a large number of requests are sent to a server as shown in Figure 1. Techniques for addressing this category of concurrent testing using TTCN-3 are presented in [8].
- Testing the processing logic of the SUT when confronted by several requests from parallel users where the state of the SUT is changing as a result of requests of the users and thus affecting each user's behavior.

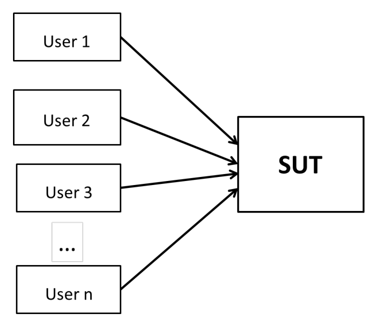


Figure 1. Parallel system configuration

In this paper, a case study, using the formal test specification language TTCN-3[9], illustrates the challenges for test coordination for this category of testing, especially with respect to race conditions, and proposes techniques to address them. We also propose shared variables and semaphores in the TTCN-3 parallel test component model as a mechanism to implement dynamic test oracles. Overall, the motivation to use a formal test specification language such as TTCN-3 and its related available execution tools to take full advantage of its logging information in order to rapidly detect faults due to race conditions. We also propose enhancements to the TTCN-3 language by introducing semaphores as a built-in language feature to make our testing concurrency problem statement usable.

There are two kinds of race conditions:

- Shared variables racing
- Behavior racing

A. *Shared variables racing*

The only way to predict a state of a SUT when multiple users are present is to use a shared variable to record for example the state of the inventory of a given product each user is trying to order. Each time a user orders a product, the inventory is decreased by one unit. However, in concurrency, a user trying to make a decision on the state of the SUT may have the shared variable overwritten by another user, thus producing a wrong result. Here the solution would be to lock the shared variable so that no other user can overwrite it.

B. *Behavior racing*

We have determined that shared variable locking is actually not sufficient to solve the race condition problem. We determined through experimentation that some sequence of events must also be locked. For instance, once the expected state of the SUT is determined using the shared variable, we must prevent another user to place a request for the same product to the SUT. The SUT also must decrease its inventory of the product. Thus if another user places an order while executing the actual purchase of a product, this will decrease the inventory and the calculated inventory by the first user will be out of synch with the real inventory of the SUT once the second user has interfered.

II. A Case Study

In sub-section A we define the dynamic state problem to be addressed. In sub-section B we propose three methods to specify concurrent systems tests.

A. *Defining the problem*

Although we have studied extensively testing of concurrency problems in industrial applications [10][11], the following simplified case study is about testing the transition of the state of a system and the kind of responses it should reply with. Here we have parallel users that send a request to a book ordering system and get two kinds of replies depending on the two possible states of the SUT: an invoice and shipping details if it has stock; or an out of stock notification. The problem is that it is impossible to predict the test oracle (predicted response) since each user is independent from each other and thus does not know the state of the SUT individually. This is similar in e-commerce applications like on-line ordering of merchandise and hotel booking and train or airline reservations systems. A typical warning message for a hotel reservation system is to warn the customer that there is only one room left at a given rate. Thus from a tester point of view, it is hard to predict if a response corresponds to a success or a failure. However, if the users are coordinated, the response to a given user can be predicted.

The interesting aspect of this simple example is that we have tried various approaches to coordination and some resulted in race condition problems, thus disturbing the test process altogether. Table I shows the values of test oracles depending on the state of the SUT, in our case: has stock; or out of stock. In short, a test passes if an invoice and shipping confirmation is received when there is inventory left or when out-of-stock is received and the server is out of stock. All other cases are failures.

Most testers use unit testing that is simple but misses some aspects of the test. Unit testing would consist of putting the SUT in the appropriate state and check the individual responses to a request.

What is missing from a unit test is the dynamic aspect of seeing the state change as the maximum available inventory is reached.

TABLE I. EXPECTED TEST ORACLES DEPENDING ON THE STATE OF THE SUT

Response to the User/state	Has stock	Out of stock
Invoice	Pass	Fail
Out of stock	Fail	Pass

B. *TTCN-3 test case implementation*

The TTCN-3 implementation of the user parallel test component (PTC) is based on a simple request/response behavior pattern with the response being analyzed with the four possible configurations of two states and two corresponding responses making use of the TTCN-3 *alt* (alternative) construct. Each alternative is guarded with the predicted state of the SUT. The receive statement contains what the received message from the SUT should match and the predicate between square brackets, the predicted state of the SUT. In Figure 2, the state variable must be provided. This happens either as setting its value while starting the PTC or computing it as the PTCs place their orders.

```
function ptcBehavior() runs on PTCType
{
  p.send("purchase");

  alt {
     [state == "has_stock"]
        p.receive("invoice") {
            setverdict(pass);
        }
      [state == "out_of_stock"]
        p.receive("invoice") {
            setverdict(fail);
        }
      [state == "out_of_stock"]
         p.receive("out_of_stock") {
            setverdict(pass);
         }
      [state == "has_stock"]
         p.receive("out_of_stock") {
            setverdict(fail);
         }
  };
}
```

Figure 2. PTC Client test verdicts situations

Instead, unit testing would break down the problem into two separate test cases and especially without the need for PTCs. Here the unit is represented by a given state.

First unit test case:

```
function unitTestBehavior_1() runs on
                          MTCType {
  p.send("purchase");

  alt {
     [] p.receive("invoice") {
          setverdict(pass);
        }
     [] p.receive("invoice") {
          setverdict(fail);
        }
```

Second unit test case:

```
     [] p.receive("out_of_stock") {
          setverdict(pass);
     }
     [] p.receive("out_of_stock") {
          setverdict(fail);
     }
   }
```

The predicates are empty because the state is predictable due to the manipulation of the SUT by the tester by emptying the data base in the first case and populating the database in the second case. Another drawback of unit testing is that the testing process would not be entirely automated since it requires a manual intervention of the tester between the two states.

Assuming that the SUT has three books on hand, the ideal testing results would be to get an invoice response for the first three users and an out of stock response for the remaining users as shown on Figure 3 and an overall pass verdict for the test.

However, the results shown in Figure 3 are only ideal and rarely happen. Instead, we see more results of the kind of Figure 4 that show the full effect of race conditions because each PTC starts at different times.

The failures shown in Figure 4 are the result of mismatches between expected and received messages when tests are executed without coordination.

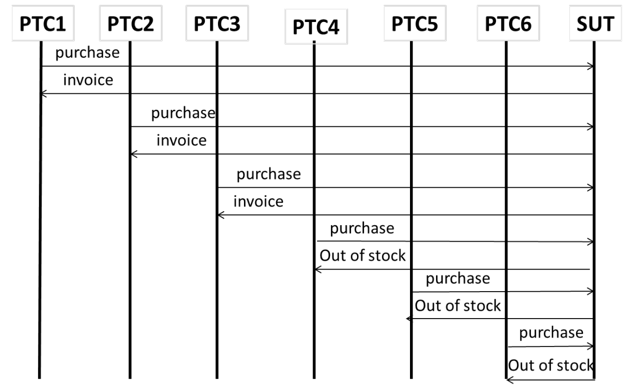


Figure 3. Ideal testing responses

This enables the tester to locate rapidly the point of failure and investigate the problem rapidly.

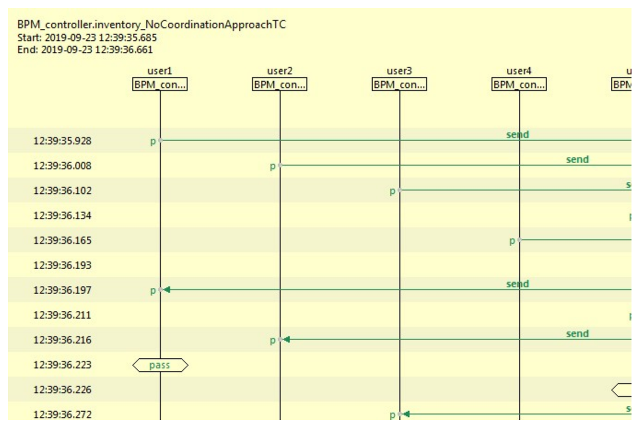


Figure 4. Uncoordinated execution results

Figure 5 shows the TTCN-3 tools data inspection feature that provides detailed message and test oracle contents that enable the tester to understand the reasons for failure especially when complex data types with many fields are used.

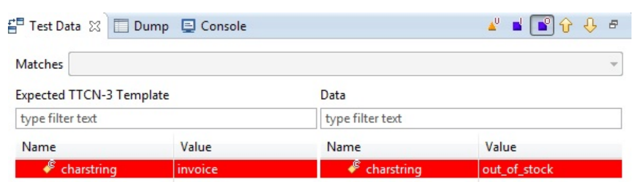


Figure 5. Expected vs received values

In this case, one may wonder where the state value comes from. This is where the test coordination is taking place. TTCN-3 has the concept of main test component (MTC) that precisely looks after that.

In our case the coordination is achieved via abstract coordination ports *cp* that link the master test component and the PTCs as shown in Figure 6.

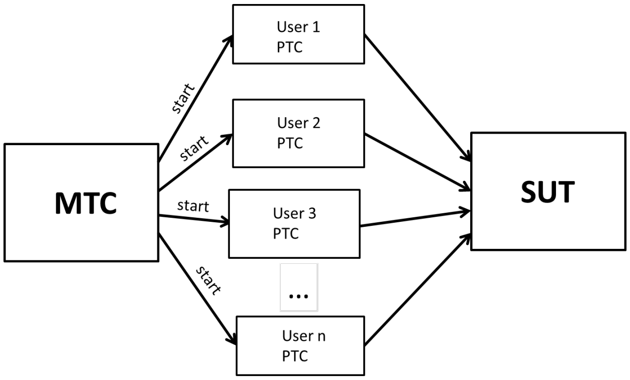


Figure 6. Test coordination with MTC

There are three ways to address test coordination.

1) Using coordination messages

The approach consists in using coordination messages between the MTC and the PTCs that contain the predicted state of the SUT. On the user PTC's side we need an additional line that receives the state from the MTC before the user attempts to test the SUT:

```
cp.receive(charstring:?)->value state;
```

On the MTC side, we send a message containing the state to the PTC that the tester thinks the server is supposed to be in. In our case this is achieved by changing the state once three requests have been placed as follows:

```
testcase coordinated_msgs_test()
   runs on MTCType system SystemType {
   …
   cp.send("has_stock") to user1;
   cp.receive("ack") from user1;

   cp.send("has_stock") to user2;
   cp.receive("ack") from user2;

   cp.send("has_stock") to user3;
   cp.receive("ack") from user3;

   // after three purchase requests,
   // the item is now out of stock

   cp.send("out_of_stock") to user4;
   cp.receive("ack") from user4;

   cp.send("out_of_stock") to user5;
   cp.receive("ack") from user5;

   …
}
```

Figure 7. Test coordination by MTC

In TTCN-3, the *receive* statement is blocking. Thus, the rest of the behavior of the PTC will not execute while the coordination message has not been received.

Note the returned *ack* message. The *ack* is used to prevent behavior racing. In other words, a new individual test cannot occur before the previous test has fully completed, otherwise more requests are being sent to the server which may change its state before a response is sent back to a user resulting in failure. We have observed that removing the ack effectively produces race conditions. We leave this verification as an exercise for the reader.

2) Coordination using PTC Threads operations

PTCs are in fact translated by the TTCN-3 compiler that produces an executable in a general purpose language (GPL) such as Java or C++ and many others using threads. Thus, one typical thread operation that is available in TTCN-3 is to check if the thread has terminated. This is represented in TTCN-3 with the keyword *done*. Here, as shown in Figure 8, each PTC is started using a parameter representing the function behavior that carries the predicted state of the SUT.

There are in fact two ways to use this feature: the first one consists in placing the done statement immediately after the corresponding start statement. This would result in transforming a concurrent system into a sequential execution system with effects similar to the coordination messages solution shown in the previous section.

```
testcase thread_operations_test()
   runs on MTCType system SystemType {
   …
   user1.start(purchasingBehavior
                      ("has_stock"));
   user2.start(purchasingBehavior
                      ("has_stock"));
   user3.start(purchasingBehavior
                      ("has_stock"));

   user1.done;
   user2.done;
   user3.done;

   user4.start(purchasingBehavior
                   ("out_of_stock"));
   user5.start(purchasingBehavior
                   ("out_of_stock"));
   user6.start(purchasingBehavior
                   ("out_of_stock"));

   user4.done;
   user5.done;
   … }
```

Figure 8. MTC behavior using PTC threads operations

In this second approach, we have chosen to place all the done statements after all the start statements for the first three PTCs to simulate the database reaching the zero inventory point. This has the advantage to at least conserve some of the concurrent behavior of the system and thus avoiding a full sequential test execution of PTCs.

3) Introducing semaphores to TTCN-3

In a way the second approach is less sequential than the first one but still somewhat sequential. Thus, we have explored a third solution that would eliminate some aspects of the sequential aspect of this test behavior. The method consists in using shared variables and Java semaphores among PTCs. The shared variable keeps track of the inventory on hand and enables a PTC to determine the state of the SUT on its own. However, TTCN-3 does not have the concept of shared variables, nor semaphores. We have explored how we could implement semaphores in the Spirent TTworkbench tool [12] using the TTCN-3 concept of external functions to link the TTCN-3 abstract behavior description to functions that are written in a GPL (e.g., Java in our case). This avoids changing the ETSI standard which would require a lengthy approval process.

III. Implementing Shared Variables and Semaphores in TTCN-3

In Java, shared variables are implemented using independent classes that contain the shared variable as an attribute. In TTCN-3, there is a simple way to reproduce this model using TTCN-3 user defined data types that allow the creation of PTCs that are actually translated into Java classes by the compiler.

For semaphores, we will use two different types of approaches:

- A semaphore data type related to an external function written in Java that creates an instance of the Java Semaphore class.
- Our own definition of semaphores directly in TTCN-3.

The second approach is for the purpose of using a more flexible kind of semaphore as opposed to the Java semaphore that blocks any process that did not acquire the semaphore. The second approach allows blocking individual resources rather than processes (here PTCs).

A. implementing shared variables

This is achieved by creating a datatype to keep track of the inventory. In this case we are trying to keep track of the inventory of two different products, product_A and product_B.

```
type component InventoryCompType
{
   var integer inventory_A := 3;
   var integer inventory_B := 4;
}
```

This data type is then used to create an instance of a PTC that will receive requests from users, compute the state of the inventory and reply with that state. Therefore we need to add a communication port that will be used to receive or send messages with the users:

```
port CoordPortType ip;
```

The basic inventory component is found on Figure 9. It is composed of two groups of events:

- Receive a request from a user
- Compute the actual state of the inventory
- Return that state to the user

```
function InventoryBehavior()
         runs on InventoryCompType {

var InventoryCompType user;

alt {
  [] ip.receive("purchase_A")
                 -> sender user {

     if (inventory_A > 0) {
      ip.send("has_stock") to user;

      inventory_A := inventory_A - 1;
     }
     else {
      ip.send("out_of_stock") to user;
     }
     repeat;
  }
  [] ip.receive("purchase_B")
                 -> sender user {

     if (inventory_B > 0) {
      ip.send("has_stock") to user;
      inventory_B := inventory_B - 1;
     }
     else {
      ip.send("out_of_stock") to user;
     }
     repeat;
  }
  [] api.receive("stop") {
     setverdict(pass);
  }
 }
```

Figure 9. Inventory computation behavior

B. *Test configuration*

There are four basic types of parallel test components as shown in Figure 10:

- The master test component (MTC)
- The inventory component
- The users components
- The System Under Test component (SUT)

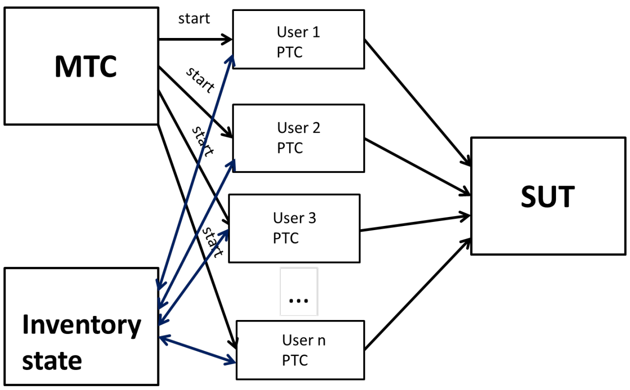


Figure 10. Shared variable handled in a separate PTC

The SUT component defined as follows:

```
type component SUTType {
      port MyPortType up;
      port MyAdminPortType aps;
      port MyAdminPortType api;
}
```

The port *up* will carry all the communication messages between the SUT and the individual user PTCs. The admin port *aps* is used at the very end of the test in order to terminate the SUT that otherwise is in a constant loop trying to receive messages from the users. The port api is used to terminate the inventory PTC that is in a similar continuous loop.

The user's component is defined as follows:

```
type component PTCType {
      port MyPortType sp;
      port InventoryPortType ip;
}
```

The port *sp* is used to communicate with the SUT while he port *ip* is used to communicate with the inventory PTC to inquire about the state of the inventory.

Finally, we define the master test component (MTC) as follows:

```
type component MTCType {
      port MyAdminPortType aps;
      port MyAdminPortType api;
}
```

The port *aps* is used to communicate with the SUT while the port *api* is used to communicate with the inventory PTC.

C. *Testcase configuration under Java semaphores using TTCN-3 external functions*

A TTCN-3 test case in our particular PTC configuration is shown in the following TTCN-3 code. As shown in Figure 11, it consists of creating an instance of a Semaphore object in the MTC that is passed on to each created user PTC and connections between the MTC and each instance of user PTCs. This enables a user to acquire or release that centralized Semaphore object.

```
Testcase
   inventory_semaphore_approach()
   runs on MTCType system SystemType {

   var Users user;
   var SemaphoreType semaphore;

   semaphore := Semaphore.new();
```

This instance of Semaphore is then passed on to each user PTC when starting their behavior function.

```
   for (var integer i:=0; i <nb_users;
                                i:=i + 1) {
      user[i].start(
          purchasingBehaviorSemaphore(
                                semaphore))
   }
```

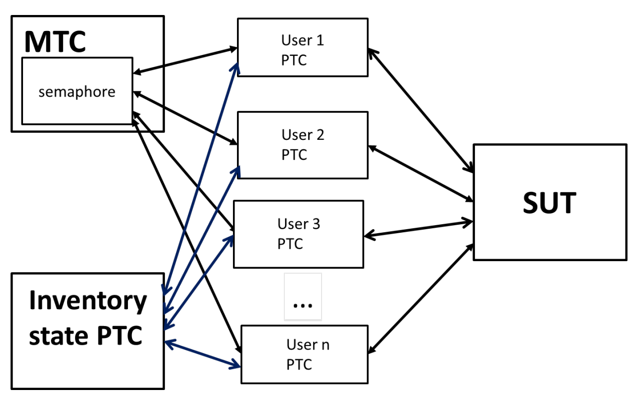


Figure 11. Java semaphore integration

In creating a SUT PTC and each user PTC and starting them using a specific behavior function, we also create an inventory PTC that keeps track of the inventory level for each product and that is connected to each PTC that query it.

TTCN-3 is based on strong typing. Thus, while the TTCN-3 source code is converted to Java, semaphores in TTCN-3 need to be typed. The translator then maps the two versions automatically. The definition of TTCN-3 semaphore is based entirely on external functions definitions as follows:

First the abstract definition of semaphores:

```
module Semaphore {
  type integer SemaphoreType;

  external function new()
        return SemaphoreType;

  external function acquire
            (SemaphoreType semaphore);

  external function release
            (SemaphoreType semaphore);
}
```

We have explored the approach of using the existing external functions definition feature of TTCN-3 to link the above abstract semaphore to the underlying Java semaphores as shown in Figure 12.

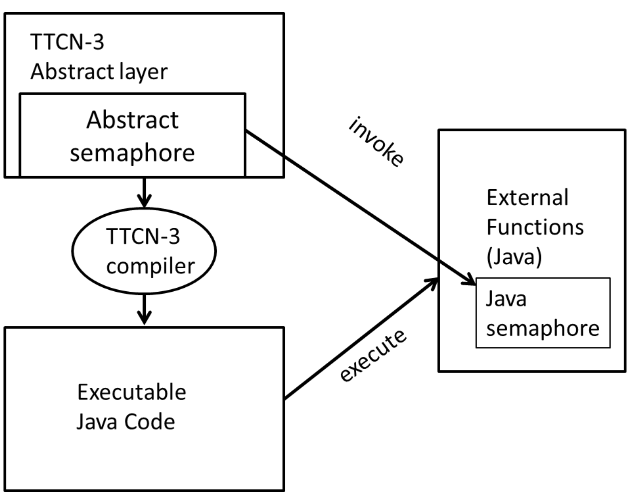


Figure 12. TTCN-3 external functions

The external function Java code is as follows:

```
package com.spirent.externalfunctions.
                            semaphore;
import java.util.HashMap;
import java.util.concurrent.Semaphore;
import org.etsi.ttcn.tri.TriStatus;
import com.testingtech.ttcn.annotation
                .ExternalFunction;
import com.testingtech.ttcn.tri.
    AnnotationsExternalFunctionPlugin;

  @ExternalFunction.Definitions(
                ExtSemaphore.class)
  public class ExtSemaphore extends
   AnnotationsExternalFunctionPlugin {

   private HashMap<Integer, Semaphore>
         semaphores = new HashMap<>();

   @ExternalFunction(name = "new",
                 module = "Semaphore")
   public int newSemaphore() {
     int result = semaphores.size();
     semaphores.put(result,
                 new Semaphore(1));
     return result;
   }

   // external function acquire(
            SemaphoreType semaphore);
   @ExternalFunction(name = "acquire",
                 module = "Semaphore")
   public void acquire(int semaphore)
   {
      try {

        semaphores.get(semaphore).
                          acquire();
      } catch (InterruptedException e)
        {throw new
                RuntimeException(e);
        }
    }

    // external function release
           (SemaphoreType semaphore);
   @ExternalFunction(name =
      "release", module = "Semaphore")
   public void release(int semaphore)
   {
      semaphores.get(semaphore).
                          release();
   }

   @Override
   public TriStatus tearDown() {
      semaphores.clear();
      return super.tearDown();
   }
}
```

While executing the test case based on external Java Semaphores, we have noticed that the results look like another type of sequence comparable to our first approach shown on Figure 3, with the only difference a random order of execution of user PTCs behavior and the calculation of the inventory state rather than the hardcoded state used in this first approach.

The user behavior is merely framed by the invocation of the semaphore acquire and release statements as shown in Figure 13.

```
function purchasingBehaviorSemaphore(
              SemaphoreType semaphore)
                     runs on PTCType {
  var charstring state;

  timer t := rnd();
  t.start;
  t.timeout;

  Semaphore.acquire(semaphore);

  //  communication with inventory PTC

  ip.send("purchase_A");
  ip.receive(charstring:?) ->
                        value state;

  // communication with SUT

  sp.send(request_product_A_t);

  alt {
    [state == "has_stock"]
     sp.receive(myInvoiceResponse_t) {
     setverdict(pass);
    }
    [state == "out_of_stock"]
     sp.receive(myInvoiceResponse_t) {
     setverdict(fail);
   }
   [state == "out_of_stock"]
  sp.receive(myOutOfStockResponse_t) {
     setverdict(pass);
  }
  [state == "has_stock"]
  sp.receive(myOutOfStockResponse_t) {
     setverdict(fail);
  }
 };

 Semaphore.release(semaphore);
}
```

Figure 13. User behavior with Java semaphore

D. Semaphores defined in TTCN-3

Effectively, the Java semaphore blocks completely the execution of user PTCs that didn't acquire the semaphore as of yet. Here we have determined that this does not provide a full concurrency behavior. Thus, instead, we have focused on a resource-oriented semaphore that would block the inventory rather than the user processes. This would allow a user to purchase product A while another user would purchase product B. There are no race condition in such a case.

In this approach, we need a mechanism in the inventory PTC to block its access by any PTC that did not successfully acquire the semaphore.

The test case is similar to the one used for the Java semaphore example with one major difference, there is no instance of a Java semaphore. Our own semaphore is actually implemented in the Inventory behavior where the inventory shared variables are protected.

1) User behavior

The user no longer uses the Java semaphore but instead invokes our own semaphore using TTCN-3 procedure invocations (call acquire) as shown on Figure 14. Since TTCN-3 procedures are non-blocking, the semaphore call will wait until the resource (inventory) is released again. This is indicated by the getreply statement that is triggered only if the call to acquire is accepted. The rest of the user behavior is identical to the one used for Java semaphores.

```
Function
     purchasingBehaviorInventoryComp()
                runs on InventoryPTC {
  var charstring state;

  semaphore.call(acquire:{}, 100.0) {
      [] semaphore.getreply {}
      [] semaphore.catch(timeout) {
            setverdict(inconc, "could
              not acquire semaphore");
            return;
      }
  }

  inventory.send("purchase_A");
  inventory.receive(charstring:?) ->
                          value state;
```

```
  p.send(request_product_A_t);
  alt {
   [state == "has_stock"]
     p.receive(myInvoiceResponse_t) {
      setverdict(pass);
     }
   [state == "out_of_stock"]
     p.receive(myInvoiceResponse_t) {
      setverdict(fail);
     }
   [state == "out_of_stock"]
     p.receive(myOutOfStockResponse_t)
                                    {
      setverdict(pass);
     }
   [state == "has_stock"]
      p.receive
           (myOutOfStockResponse_t) {
        setverdict(fail);
      }
   }
 }

 semaphore.call(release:{});
}
```

Figure 14. user behavior with our definition of semaphores

2) Use of semaphores in the inventory PTC

The procedures acquire_A and release_A and their corresponding names for product B are residing in the inventory behavior. Here we are using the variables inventory_A_blocked and its counterpart for B to control which part of the inventory calculation code can be reached.

```
function InventoryBehavior()
          runs on SemaphoreInventory {
  var integer blockedBy_A := -1;
  var integer blockedBy_B := -1;
  var InventoryPTC user;
  var boolean inventory_A_blocked :=
                                false;
  var boolean inventory_B_blocked :=
                                false;

  alt {
   [inventory_A_blocked == false] any
     from sem_p.getcall(acquire_A:{})
         -> @index value blockedBy_A {

       inventory_A_blocked := true;

       sem_p[blockedBy_A].reply(
                acquire_A:{});
       repeat;
   }
   [inventory_A_blocked == true]
       sem_p[blockedBy_A].getcall(
                    release_A:{}) {

     blockedBy_A := -1;

     inventory_A_blocked := false;
     repeat;
   }
   [inventory_B_blocked == false] any
     from sem_p.getcall(acquire_B:{})
        -> @index value blockedBy_B {

     inventory_B_blocked := true;
     sem_p[blockedBy_B].reply(
                     acquire_B:{});
                       repeat;
   }
   [inventory_B_blocked == true]
         sem_p[blockedBy_B].getcall(
                     release_B:{}) {
     blockedBy_B := -1;
     inventory_B_blocked := false;
     repeat;
   }

   [inventory_A_blocked == true]
    inventoryPort.receive(
     "purchase_A") -> sender user {
     if (inventory_A > 0) {
       inventoryPort.send(
             "has_stock") to user;

       inventory_A := inventory_A -
1;
     }
     else {
        inventoryPort.send(
           "out_of_stock") to user;

          repeat;
     }
   [inventory_B_blocked]
     inventoryPort.receive(
       "purchase_B") -> sender user {

     if (inventory_B > 0) {
      inventoryPort.send("has_stock")
                            to user;

      inventory_B := inventory_B - 1;
     }
     else {

      inventoryPort.send(
                    "out_of_stock")
                            to user;
```

```
      }
      repeat;
    }
    []  api.receive("stop") {
        setverdict(pass);
    }

  }
}
```

Figure 15. Inventory behavior with TTCN-3 semaphores

The execution of a single component's behavior produces the sequence of events shown on Figure 16. We can clearly observe that when the inventory PTC replies with the *has_stock* message, the actual purchase message sent to the SUT results in an invoice coming back from the SUT.

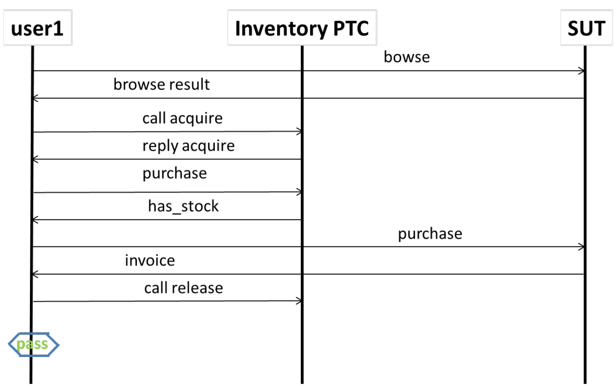


Figure 16: Single user sequence of events

E. Mutually Exclusive Behavior Blocks

As can be seen easily, this way of modeling semaphores is very complicated and the code obfuscates the intent, reducing the manageability.

We therefore propose to add a new feature to TTCN-3 that allows controlled, shared access to component variables from the scope of other component behaviors.

This would be syntactically modelled by an on-statement of the following form:

on *ComponentRef* [**@readonly**] *StatementBlock*

The *StatementBlock* would be allowed to access component variables of the component referenced via *ComponentRef* via dotted notation.

The on-statement uses an implicit read lock (in case of @readonly) or read-write lock behavior otherwise of a re-entrant lock associated with the referenced component. It is assumed that there is one such re-entrant execution lock associated with every created component. If the read-write lock is acquired by any component, no other component can acquire the lock of that component. If the read-lock of a component is acquired by any component, the read-write lock of the component can not be acquired.

Components trying to acquire a lock that can currently not be acquired block their execution until the lock becomes available again or their behavior is terminated.

The read-write lock is acquired whenever the component starts executing behavior or when a non-read only on-statement for that component is entered. It is released when the component terminates its current behavior or starts waiting in an alt statement or when the outermost on-statement referencing the component in the behavior of another component is left.

The read-lock of a component is acquired whenever a read-only on-statement for that component is entered. Thus, it is possible that several components read the component variables of one component in parallel or that a single component has read-write access to the referenced components variables at any one time.

The advantage of this more declarative approach is that the code is more readable and that it is possible to statically analyse the types and names of the used variables, whether they are only read in read only on-statements.

Using this construct, the inventory component becomes a simple shared variable container without any executed behavior and is passed as a reference to each PTC for referencing when starting the following behavior.

```
function purchasingBehaviorInventory
(InventoryComp inventory)
runs on PTCType {
  on inventory {
  // acquired read-write lock of
  // inventory
    var charstring state;
    if (inventory.inventory > 0) {
      state := "has_stock";
      inventory.inventory :=
        inventory.inventory - 1;
    } else {
      state := "out_of_stock";
    }
    p.send(myRequest_t);
    alt {
    [state == "has_stock"]
      p.receive(myInvoiceResponse_t) {
        setverdict(pass);
      }
    [state == "out_of_stock"]
```

```
        p.receive(myInvoiceResponse_t) {
          setverdict(fail);
        }
      [state == "out_of_stock"]
    p.receive(myOutOfStockResponse_t) {
          setverdict(pass);
        }
      [state == "has_stock"]

    p.receive(myOutOfStockResponse_t) {
          setverdict(fail);
    }
      }
  } // release read-write lock of
    // inventory
}
```

F. *Mixing blocked and free behavior*

Testing web applications has been studied intensively for several decades. An early attempt can be found in [14]. However, none of them mention race conditions.

So far, we have shown the simple behavior of users that create race conditions. This is achieved by blocking shared variables and behavior sequences. However, events in an e-commerce application are not always subject to race condition. The race condition occurs only when a user is trying to make a purchase and thus the state of the inventory must be determined. On the other hand, browse events are not subject to race conditions at all. They can occur any time without affecting inventories states. Thus, we have created an example where browse events precede the purchasing event and we have introduced sequences of different purchase events.

With several users and the alternate blocked and unblocked behaviors of users we may end up with various sequences of events interleaving between the users. The sequence in Figure 17 shows that user2 places its order of product A before user1 places its order for product B. The actual specification of a user's behavior may suggest that a given user may place orders for product A and product b in strict sequence without interruption by another user but this figure clearly shows that this is not the case. Here the unblocked browse event allowed such an interleaved sequence. However, it is to be noted that block portions of behavior remain together and thus cannot be interrupted by another user.

IV. TTCN-3 AS A MODELLING LANGUAGE

Normally, testing activities can take place only once the SUT has been fully developed and is runnable. However, planning and developing automated test cases can be done in parallel to the SUT development phase. More importantly, the missing SUT can be emulated using TTCN-3. This enables us to find any flaws in the automated test suites before we apply them to the SUT and thus reduce time to market.

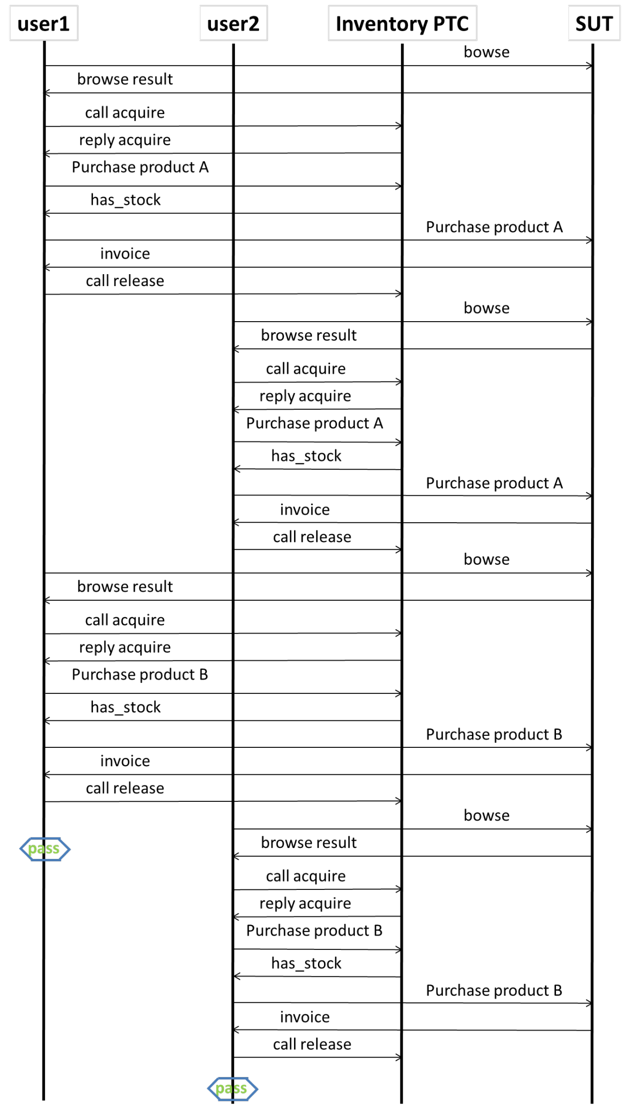


Figure 17. Multiple users: possible sequence of events

In our case study, this means finding a way to portray a behavior that replies with "invoice" when there is inventory on hand and replies "out of stock" when inventory has reached zero. At the abstract level, there is no need to implement a full system, in our case probably a web application and a related database. The implementation of such an abstract system is as follows:

```
function SUTbehavior() runs on SUTType
{
   var integer inventory := 3;
   var PTCType ptc := null;
   var MTCType mtc_ := null;

   alt {
      [] p.receive("purchase") ->
```

```
                        sender ptc {
            if(inventory > 0) {
               p.send("invoice") to ptc;
               inventory := inventory -1;
            }
            else {
               p.send("out_of_stock") to
                                       ptc;
            };
            repeat
         }
         [] ap.receive("stop")
                  -> sender mtc_
               setverdict(pass)
         }
      }
   }
```

Figure 18. SUT behavior

We use a simple variable to portray the inventory that we set at 3 units. Every time a request to purchase an item comes in, we decrease the inventory. A simple if-then-else statement provides the correct response of *invoice* or *out-of-stock* state. At the abstract level, this is all we need.

Also, the test suite is developed in two different levels of abstraction. First, we use simplified messages like simple strings. Once we simulate the abstract system and we are happy with the results, in a second step we merely redefine the abstract data types and its corresponding templates (test oracles for received messages and data content for sent messages) as follows:

1st step: Data types and templates declarations:

```
type charstring RequestType;
type charstring ResponseType;

template RequestType myRequest_t :=
                        "purchase";

template ResponseType
            myInvoiceResponse_t
                        := "invoice";
template ResponseType
            myOutOfStockResponse_t:=
                     "out_of_stock";
```

Figure 19. Simplified data types and templates

2nd step: Real data types and templates:

```
type record RequestType {
    charstring bookName,
    charstring ISBN
}

type record ResponseType {
    charstring bookName,
    charstring ISBN,
    charstring status,
    charstring action
}

template RequestType myRequest_t := {
    bookName := "ttcn-3 in a
nutshell",
    ISBN := "978-2-345-678"
}

Template ResponseType myResponse_t :=
{
      bookName := "war and peace",
      ISBN := "978-2-345-678",
      Status := "on hand",
      Action := "invoice"
}
```

Figure 20. Fully realistic data types and templates

Note that both datatypes and templates are defined using the same identifiers. Only their content is different.

V. Conclusion

Despite its long history, testing concurrent systems remains complex and does not always provide accurate results. In this paper we have shown that using formal methods for testing such as TTCN-3 helps to locate problems accurately because of the wide choice of results visualization features that the various commercial and open source editing, and execution tools provide. We also have experimented with the TTCN-3 external functions concept in order to implement shared variables and Java semaphore features for the MTC and the PTCs. We have discovered that the traditional Java semaphores approach prevents true concurrency testing and we have developed a system that allows blocking resources rather than processes to avoid racing conditions which provides a considerably more flexible concurrency testing.

Acknowledgment

The authors would like to thank NSERC for funding this research.

References

[1] B. Stepien and L, Peyton, Test Coordination and Dynamic Test Oracles for Testing Concurrent Systems, in Proceedings of SOFTENG 2020

[2] E. Boros and T. Unluyurt, Sequential Testing of Series-Parallel Systems of Small Depth, in ISBN 978-1-4613-7062-8

[3] A. Bertolino, Software Testing Research: Achievements, Challenges, Dreams in Proceedings of FOSE '07 pp. 85-103

[4] T. Hanawa, T. Banzai, H. Koyzumi, R. Kanbayashi, T. Imada and M. Sato, Large-Scale Software Testing Environment Using Cloud Computing Technology for

Dependable Parallel and Distributed Systems in 2010 Third International Conference on Software Testing, Verification and Validation Wokshops Procedings

[5] A. M. Alghamdi and F. Eassa, Software Testing Techniques for Parallel Systems: A Survey in IJCSNS International Journal of Computer Science and Network Security, vol 19, No. 4, April 2019, pp. 176-184

[6] L. Parobek, 7 Reasons to Move to Parallel Testing in white paper on https://devops.com/7-key-reasons-make-move-sequential-parallel-testing/, last accessed December 14th, 2020

[7] B. Rao G. , K. Timmaraju, and T. Weigert, Network Element Testing Using TTCN-3: Benefits and Comparison in SDL 2005, LNCS 3530, pp. 265–280, 2005

[8] G. Din, S. Tolea, and I. Schieferdecker, Distributed Load Test with TTCN-3, in Testcom 2006 Proceedings, pp. 177-196

[9] ETSI ES 201 873-1, The Testing and Test Control Notation version 3 Part 1: TTCN-3 Core Language, May 2017. Accessed March 2018 at http://www.etsi.org/deliver/etsi_es/201800_201899/20187301/04.09.01_60/es_20187301v040901p.pdf

[10] B. Stepien, K, Mallur, L. Peyton, Testing Business Processes Using TTCN-3, in SDL Forum 2015 proceedings, Lecture Notes in Computer Science, vol 9369. Springer, Cham. Pp. 252-267

[11] B. Stepien, L. Peyton, M. Shang and T. Vassiliou-Gioles, An Integrated TTCN-3 test framework architecture for interconnected object-based applications in IJEB vol. 11, No. 1, 2014

[12] TTworkbench,Spirent, https://www.spirent.com/Products/TTworkbench, last accessed December 14th, 2020

[13] Titan, https://projects.eclipse.org/proposals/titan, last accessed December 14th, 2020

[14] F. Ricca and P. Tonella, Testing Processes of Web Applications in Annals of Software Engineering 14, pp. 93-114 (2002)

Reusing and Deriving Games for Teaching Software Reuse

Diego Castro, Cláudia Werner

COPPE/Computer Systems Engineering Program
Federal University of Rio de Janeiro
{diegocbcastro, werner}@cos.ufrj.br

Rio de Janeiro, Brazil

***Abstract*—Software Reuse (SR) is a crucial discipline that seeks to create new products using pre-existing artifacts. Companies are increasingly looking for members with these skills. However, they do not always find them. One of the possible causes is the lack of dedication of students in the classroom. To increase students' motivation and engagement in the classroom, many scholars are already using games as a teaching method, due to the various advantages that this strategy can bring to the current teaching method. SR can be related to several areas, such as Software Engineering, processes, or programming, more commonly associated with the latter. Based on this, the paper sought to identify games that aimed to teach programming but could also be used to teach the fundamentals of software reuse. Two games were created/derived from those found, and it was concluded that games could be derived or reused for more than one teaching context.**

Keywords–game; game-based-learning; software reuse; programming; systematic mapping.

I. INTRODUCTION

This work is an extension of the paper presented by Castro and Werner [1] at the 6th International Conference on Advances and Trends in Software Engineering (SOFTENG - IARIA), Lisboa, Portugal.

In general, software is still largely built from scratch. Ideally, existing documents (source code, project documents, etc.) could be copied and adapted to new requirements. However, the goal of reuse is still far from ideal [2]. Reuse is something intrinsic to people, and nothing is made from scratch - everything is built from something [3]. This is the basic principle of reuse, creating something from something previously built.

Software Reuse (SR) is a discipline based on this context, seeking to use existing elements (which were built to be reused) to create new ones. In general, the term SR refers to a situation in which some software is used in more than one project. In this context, software is defined as one or more items that are part of a productive organization's process. Thus, software can refer to the source code or any other product in its life cycle, such as requirements, designs, test plans, documentation, etc. [4]. With the correct use of this discipline, it can provide several positive impacts in a variety of contexts, such as quality, cost, productivity, code-making performance, rapid prototyping, reduced code writing, reliability, and interoperability of software [2].

Despite the advantages offered by reuse, discomfort is usually expressed by statements such as: "We don't want to do extra work to benefit someone else"; "We can do it better"; "We won't use it if it was created by someone else"; and even competition within organizations can be an obstacle in this regard [2]. Despite all these problems mentioned, there is still one that is seen as the main factor for reuse not being implemented, which is the difficulty of learning in this area [5].

One of the biggest obstacles that educators face in the teaching process is the need to use methods that effectively make students pay attention in class and learn easily. Many students with little class time have already lost their focus in the discipline being taught. This can be a reflection of the passive teaching method being strongly centered on the teacher, lectures, and slides without approaches that capture the student's attention [6].

Based on this, educators are increasingly looking for innovative learning strategies that combine pleasure with education so that it is possible to solve problems by teaching a subject [6]. Thus, alternative teaching methods are sought with which the student can interact and make better use of teaching, in a practical and attractive way. Based on this, a possible solution to make this learning more pleasurable is the use of games as a teaching tool. Most games are interactive, which is one of the main ways of distraction and pleasure, making them an excellent way to remind students of what has been taught [7].

The study presented in this paper is part of a more extensive study that sought to find and produce games for teaching software reuse. The study initially did not find any work that made specific reference to a game for teaching reuse. Still, it was observed that SR might be contained in different areas, such as programming, Software Engineering (SE), among others. However, the programming area is usually the most referenced [8]. Based on the information provided, this study aims to identify games that have the purpose of teaching programming with emphasis/potential for reuse, that is, to find games that were developed for teaching programming, but could be used to teach some of the fundamentals of software reuse, such as logical reasoning development, function development, object orientation, among others.

Based on each of the games found, a possible modification or its use for teaching SR was proposed. Based on these proposals, it was understood that games could be derived or reused for more than one context. With this, two more games were derived from those found for SR teaching.

The remainder of this paper is presented as follows: Section II presents some related works, Section III describes the research method used in the systematic mapping, Section IV shows some results that were found, Section V demonstrates an example of how one of the games found could be used to teach SR, and Section VI shows the threats to validity and concludes with the final remarks.

II. Related works

This section aims to briefly present some related works that served as inspiration for this study. The papers show studies on teaching programming through games and make some observations about this method.

According to Combéfis, Beresnevičius, and Dagienė [9] several game-based platforms offer programming teaching. In their work, they sought to identify several games that aim to teach programming; each game's characteristics and elements are shown in the paper.

Malliarakis et al. [10] sought to review the features that should be supported by educational games and which educational games already support these features for programming. In this work, several games were reviewed. Their characteristics were demonstrated through a relationship table, showing each of the observed games' elements and features.

Miljanovic and Bradbury [11] evaluated 49 games for teaching programming to find characteristics of friendliness, accessibility, learning effect, and involvement. From the tables of elements presented in the paper it was possible to identify several games that sought to teach concepts involved with Software Reuse, such as functions, object orientation, and recursion. It is worth remembering that the research areas demonstrated in these papers are part of both Programming and Software Reuse, and there is an intersection of content between them.

From these works and related information, it was possible to observe that many of the observed games had essential characteristics that could be used for teaching Software Reuse.

III. Research Method

Systematic mapping is a secondary study method based on a structured and repeatable process or protocol that explores studies and provides a result in the form of an overview of a particular subject [12]. The mapping presented follows the protocol proposed by Kitchenham [13].

The research process presented in this study covers articles published by the end of 2018 and aims to conduct a systematic mapping to identify games that were built for programming teaching but could be used to teach software reuse fundamentals, such as logical reasoning development, function development, object orientation, among others.

A. Research Questions

- **Q1:** What is the main advantage / motivation of the use of games to teaching programming language?
- **Q2:** What is the disadvantage of the use of games to teaching programming language?
- **Q3:** What is the main characteristic of the game used?
- **Q4:** What was the evaluation method used?

The mapping presented followed well-defined steps so that it was possible to reach a set of articles that were of interest to the search [13]. The search string was executed in Scopus as recommended by other studies [14] [15], and then the inclusion and exclusion criteria were applied to the set of articles that were found based on the title, abstract, and full text.

B. Inclusion criteria

- The article must be in the context of using games for programming language teaching;
- The article must provide clues about software reuse
- The article must provide data to answer at least one of the research questions;
- The article should be written in English.

C. Exclusion criteria

- Book chapters, conference call;
- Studies that can not be fully accessed (i.e., papers that could not be downloaded).

D. Search string and Analysis

The definition of the search string was based on the **P**opulation, **I**ntervention, **C**omparison, **O**utcome (PICO) structure [16], using three of the four levels. The search string was defined by grouping the keywords of the same domain with the logical operator "OR" and grouping the two fields with the logical operator "AND". However, we chose to use a date filter, searching only for articles that were published within five years, aiming to find more recent works in the area [17]. Table I demonstrates the PICO structure used in conjunction with the search string. Initially, the search string returned a total of 507 papers. When analyzed according to the inclusion and exclusion filters, this number dropped to 17 papers.

According to Motta et al. [14] and Matalonga et al. [15], the snowballing procedure can mitigate the lack of other search engines, complement the strategy. Therefore, to minimize the loss of some articles, the snowballing procedure were chosen, looking at the references and citations of the articles looking for relevance [15]. The snowballing process is divided into two stages, backward and forward. The backward process aims to identify new articles based on the works that were referenced in the article that was analyzed, and the forward refers to the identification of new papers based on the works that referenced the paper that was analyzed [15]. From the use of this procedure, it was possible to include nine more papers, three of them through the forward process, and six through the backward process, resulting in a total of 26 papers. Figure 2 demonstrates a summary of the data extraction process that was used in this work. Table II shows how these 26 articles were obtained, and Table III shows each of these articles, demonstrating which questions were answered by each paper.

TABLE I. SEARCH STRING

PICO	SYNONYMS
Population	Programming language, algorithm experience, algorithm skills, algorithm alternative, algorithm method, coding experience, coding skills, coding method, coding alternative
Intervention	Tutoring, teach*,instruction, discipline, schooling, education*, mentoring, course, learn*,train*, syllabus
Comparison	Not applicable
Outcome	Game*, gami*, play*, "serious games", edutainment, "game based learning", simulation
SEARCH STRING	
TITLE-ABS-KEY (("programming language" OR "algorithm experience" OR "algorithm skills" OR "algorithm alternative" OR "algorithm method" OR "coding experience" OR "coding skills" OR "coding method" OR "coding alternative") AND (tutoring OR teach* OR instruction OR discipline OR schooling OR educat* OR mentoring OR course OR learn* OR train* OR syllabus) AND (game* OR play* OR "serious games" OR gami* OR edutainment) AND (LIMIT-TO (PUBYEAR , 2018) OR LIMIT-TO (PUBYEAR , 2017) OR LIMIT-TO (PUBYEAR , 2016) OR LIMIT-TO (PUBYEAR , 2015) OR LIMIT-TO (PUBYEAR , 2014)))	

TABLE II. ANALYSIS OF THE PAPERS

	Main Study		Snowballing backward		Snowballing Forward	
Activity	Result	Number of papers	Result	Number of papers	Result	Number of papers
First Execution	507 added	507	389 added	389	123 added	123
Repeated Papers	6 withdraw	501	294 withdraw	95	16 withdraw	107
Papers in another language	0 withdraw	501	14 withdraw	81	13 withdraw	94
Remove conference / workshops	16 withdraw	485	0 withdraw	81	0 withdraw	94
Remove books	0 withdraw	485	0 withdraw	81	0 withdraw	94
Remove by title	368 withdraw	117	46 withdraw	35	58 withdraw	36
Remove by abstract	83 withdraw	34	17 withdraw	18	18 withdraw	18
Papers not found	0 withdraw	34	0 withdraw	18	0 withdraw	18
Remove by full paper	17 withdraw	17	12 withdraw	6	13 withdraw	3
Total papers included	17 papers		6 papers		3 papers	
Extracted Papers	26 papers					

TABLE III. TRACEABILITY MATRIX.

Title	Year	Q1	Q2	Q3	Q4
Perceptions of Scratch programming among secondary school students in KwaZulu-Natal, South Africa	2018	X		X	X
Robo3: A Puzzle Game to Learn Coding	2018	X	X	X	X
Improving programming skills in engineering education through problem-based game projects with Scratch	2018	X		X	X
Introducing novice programmers to functions and recursion using computer games	2018	X		X	X
Introducing programming using "scratch" and "greenfoot"	2018	X		X	X
Developing Educational 3D Games With StarLogo: The Role of Backwards Fading in the Transfer of Programming Experience	2018	X	X	X	X
Learning to think and practice computationally via a 3D simulation game	2018	X		X	X
Design and implementation of Robo3 : an applied game for teaching introductory programming	2017	X		X	X
A cross-cultural review of lightbot for introducing functions and code reuse	2017	X		X	
Using Digital Game as Compiler to Motivate C Programming Language Learning in Higher Education	2017	X		X	X
Cubely: Virtual reality block-based programming environment	2017	X		X	X
Analysis of the learning effects between text-based and visual-based beginner programming environments	2017	X		X	X
Visual programming language for model checkers based on google blockly	2017	X		X	X
Educational resource based on games for the reinforcement of engineering learning programming in mobile devices	2016	X		X	X
Teaching abstraction, function and reuse in the first class of CS1 - A lightbot experience	2016	X		X	X
From Alice to Python Introducing text-based programming in middle schools	2016	X		X	X
Visual programming languages integrated across the curriculum in elementary school: A two year case study using Scratch" in five schools	2016	X		X	X
Building a Scalable Game Engine to Teach Computer Science Languages	2015	X		X	X
A mobile-device based serious gaming approach for teaching and learning Java programming	2015	X		X	
Coding with Scratch: The design of an educational setting for Elementary pre-service teachers	2015	X		X	X
Droplet, a Blocks-based Editor for Text Code	2015	X		X	X
Integrating Droplet into Applab – Improving the usability of a blocks-based text edit	2015	X		X	X
The development of a virtual learning platform for teaching concurrent programming languages in secondary education: The use of open Sim and Scratch4OS	2014	X	X	X	X
Effects of using Alice and Scratch in an introductory programming course for corrective instruction	2014	X		X	X
A structured approach to teaching recursion using cargo-bot	2014	X		X	X
The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners, Perspective, Informatics in Education	2014	X		X	X

IV. RESULTS

The articles found in this study sought to demonstrate games that could be used in teaching some concepts related to programming. However, the analysis of the documents was performed in search of works that could be used to explain some of the concepts of SR. From this, works that were not developed with this context but could be used for this purpose were also found. Figure 2 groups the articles by location and year of publication. It is possible to see an increase in the number of publications over the years and that many countries are looking for improvements in this area.

The bottom of the figure also shows the number of articles found grouped by game type. However, some papers used more than one approach. From Figure 2, it is possible to observe that the most used way to teach programming is through the use of ”blocks of code”. Although this is not a real game, it uses many features similar to games, such as the use of graphical interfaces, sounds, and tasks to be done. Thus, these papers were also considered in this work.

Q1: What is the main advantage / motivation of the use of games to teaching programming language?

Video games have been in our lives for more than 50 years, quickly becoming one of the most important, profitable, and influential forms of entertainment [19]. A game is an activity between two or more decision-makers who seek to achieve their goals in some limited context. A more conventional definition is that games are systems in which users participate in an artificial conflict, defined by rules, which ends in a

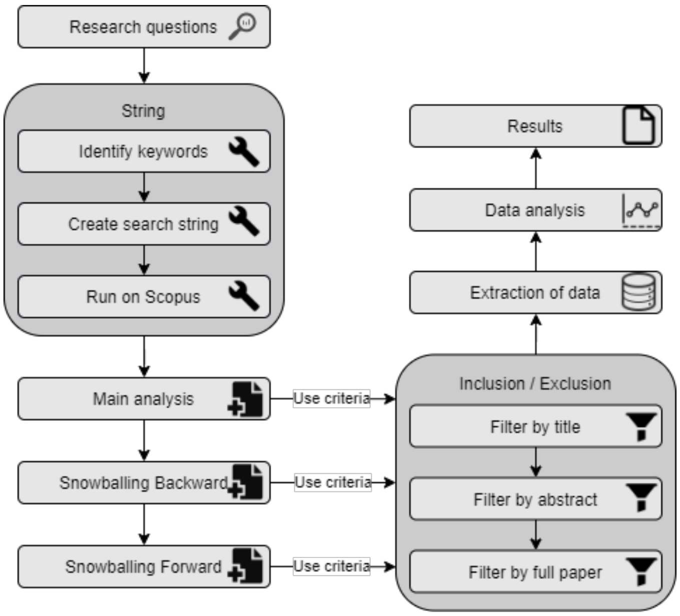


Figure 1. Summarization of the research protocol based on the model created by Calderón et al. [18]

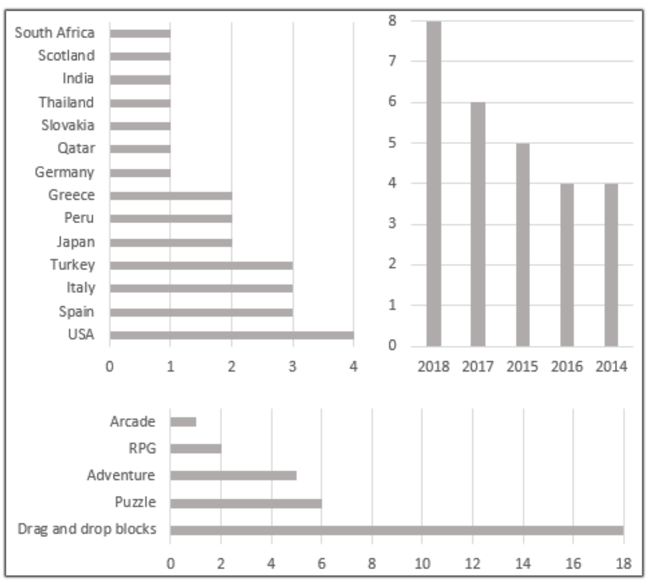


Figure 2. General analysis of the articles found.

quantifiable result [20].

Games are a visible and physical representation of a problematic space, a captured mental model that can be repeated. They are places to: test new ideas and experiment theories; repeat the training as many times as necessary; where time and space can be contracted or expanded; learn more from failure than from success [21] [19].

Many benefits are associated to the use of games in education, such as increased collaboration and competition, the creation of immediate feedback, the possibility of reflecting on the results achieved and the transfer of content so that learning is an integral part of gameplay, where the student must use the knowledge acquired in class to solve a problem within the game [22] [23] [24].

TABLE IV. Advantages of using games as a teaching technique.

Advantage	References
Entertainment	[22, 23, 27, 29–37]
Practical knowledge	[20, 23, 27, 37, 38, 38–44]
Engagement and motivation	[23, 29, 31, 35, 36]
Knowledge retention	[22, 23, 35, 36, 45]
Interactivity	[36]
Immediate feedback	[29, 31, 32, 35]
Reflection	[23, 36]
Immersion	[36]
Real Scenery	[32, 36]
Risk-free experience	[36]

Using games as a reinforcement tool to teach skills can be a very beneficial strategy for students. They have proven to be a useful tool to complement conventional learning methods. Games allow visualizing concepts that may be too abstract. They also help you get acquainted with the knowledge and methods that may be tedious to study, offering a cycle of challenges and rewards that drives the learning experience [25].

Many authors claim that games have several characteristics that can benefit teaching [22] [26]. They have already been used as successful educational tools in many different fields and topics, such as engineering, learning languages, theater and even health [27]. The advantages include: increased student motivation and engagement, enhancement of pre-existing knowledge, increased performance in practical activities, immediate feedback, fun and satisfaction, among others [25].

Finally, some visual programming languages were also identified that are not directly considered games, but that use very similar characteristics, such as increased motivation and student engagement, use of graphical interfaces, among others. Visual programming is the use of visual expressions in a programming environment as an editing shortcut to generate code that may or may not have a different syntax than that used in textual coding [28]. There are several advantages to using visual programming, and the main ones are [5] [12]: Code generation through the combination of blocks; Make programming more accessible to a specific audience; Improve the quality with which people perform programming tasks; Reduce the number of errors generated by beginning programmers, not having to worry about the language syntax; Improve the speed with which people perform programming tasks; feedback and visual execution; and Minimize the command set.

Several advantages were found in relation to the use of games as a teaching strategy, Table IV demonstrates each one of these.

Q2: What is the disadvantage of the use of games to teaching programming language?

The limited use of serious games in formal education may be related to the issues surrounding the use of leisure games, such as a view where games can be addictive, are not productive, and can teach wrong concepts. Another very relevant point regarding the non-use of games as a teaching medium is that players usually learn through repetition, patterns, and exploration, which contrasts with the learning of discrete amounts of information, as it occurs in schools and undergraduate courses in general [46].

Despite the advantages offered by games as a teaching method, there are also some issues involving this approach. The first problem found was the comparison of the level of learning provided by a game as a teaching method and a class with textual programming. Despite the advantages offered by games, textual programming can still convey better content [35]. Despite the advantages offered by visual programming, it was also observed that text-based programming achieved better results in relation to knowledge transfer [21].

Another problem identified was the complexity of the game created. If the teaching tool used is too complicated, students can reduce the time spent solving problems to focus more on the tool. This is an unwanted distraction, and any game used should be easy to use, allowing the student to focus on solving the problem rather than how to use the game [38].

Finally, the last problem identified was that although games provide several advantages, they are not seen as self-sufficient. Professional follow-up and feedback on the course are required to solve any problem that may arise throughout the learning process [38]. Therefore, it is unlikely to teach students new content just by using games. They are mostly used to reinforce content that has already been presented by a teacher.

Q3: What is the main characteristic of the game used?

This study identified several games that sought to teach programming through increased motivation and engagement through fun. Most of these games were designed to be used by users with minimal or no knowledge of programming languages [45].

The first game found was LightBot, which is a game to teach programming logic and has features such as multi-level, difficulty progression, feedback, challenges, use of similar tasks, concepts of functions, abstraction, flow control and recursion [29] [31] [34]. Another game very similar to the one described above is Cargo-Bot, which has the same characteristics, but with other gameplay that revolves around a crane that moves and stacks a set of colored boxes. Players write small programs to move boxes from one initial setup to another [23]. Another game called Robo3 was found that had characteristics very similar to those described [25].

Another game very similar to the ones listed above was a game designed to teach Java programming that, to advance the levels, the player needs to overcome different levels. As the player surpasses these levels, he or she can progress through the story, unlocking new elements and gaining experience points to unlock new content [45].

Another game found was Lost in Space, which includes, among other components, a game rules system, a physics engine, and a rendering engine. The game screen is divided into two parts. The left side containing the code interpreter text area and a help window and on the other side, the game phase. Through this game, some features were highlighted, such as obstacles, code interpreter (pseudocode of the game), collisions, movement, enemy and attack system [27].

In this research, we also identified some visual programming languages that are not considered as games directly, but that uses ”block” approach to building programs. The first two to be identified were Alice [32] and Scratch [30], which are block-based visual programming languages designed to promote media manipulation for new programmers. From these languages, it is possible to upload media projects and scripts, animated stories, games, book reports, greeting cards, music videos, tutorials, simulations, and art and music projects. Two other languages very similar to those described are StarLogo TNG [35] and Droplet [47] [48], which are also drag-and-drop visual languages.

TABLE V. Characteristics used in games as teaching techniques.

Characteristics	References
Score	[17, 23, 25, 27, 29, 31, 45, 50]
Levels	[23, 25, 27, 29, 31, 45]
Checkpoints	[23, 25, 29, 31, 45]
Competition and collaboration	[22]
Feedback	[23, 25, 27, 29, 31]
Simulation	[36]
Final result	[23, 27, 29, 31, 36, 45]
Real scenario	[36]
Challenge	[22, 23, 25, 27, 29, 31, 36, 45]
Dependence between contents	[22, 29]
Stimulating graphics	[22, 23, 27, 29, 36]

Greenfoot is an integrated tool that aims to teach object-oriented programming. Also, the tool allows teachers to introduce the most essential and fundamental concepts of object orientation in an easily understandable way [49]. Finally, the last visual language found is called Google Blockly [43], which is a library for building visual programming editors.

Finally, another feature that was used to create these games was the use of virtual and augmented reality. The Cubely game made use of these technologies to develop an idea that blended block programming concepts and the Minecraft game [36].

Several characteristics were found in the games that were evaluated. Table V shows each of them.

Q4: What was the evaluation method used?

Several evaluation methods were identified in this research. However, in general, all evaluations have a questionnaire applied to a specific population after using the tool to validate it [35] [32] [43].

Another possible means of the evaluation was the use of control groups where one group used the tool, and the other did not, and the same questionnaire was applied to both groups [27]. Through this assessment, it is possible to find out if there was a gain of experience through the tool use since it is possible to compare the results of the two groups.

The last evaluation method found was about the use of the tool as part of the discipline — the tool as a complement to the teaching of programming [42].

V. Discussion

A. *Reusing games to teach SR*

This mapping found several games; however, none of them was produced to teach SR. Nevertheless, these games, with only a few or no modifications, could be used to explain certain concepts of SR, such as the importance of reusing, software components or code reuse.

Thinking about this idea of teaching SR, visual language platforms, such as Scratch [30], Alice [32], Droplet [47], and Google Blockly [43] could, for example, be used to teach code reuse or software components. A software component can be understood as a software unit with a well-defined interface and explicitly specified dependencies. It can have different sizes and can be characterized in different ways, from a small piece of code, a software package, a web service, a module that encapsulates a set of functions, or it can even be as large as a program [2]. All the visual languages have a strong base of reuse, where the user creates programs by joining blocks (i.e., pre-produced components). Therefore, it is possible to understand the "blocks" of visual programming code as software components. Both (blocks and components) can be seen as pieces of code that exercise a specific functionality and must be reused to produce a new program.

Robo3 [25], CargoBot [23] and Lightbot [29] are games of puzzle type and are very similar. The general idea of these games is to create sequences of activities (which are described as functions) that perform a task, such as taking the avatar from point A to point B or moving boxes from one initial setup to another. Thinking about this type of games, these functions can be used in the game several times, teaching the student the concept and importance of code reuse. Cubely [36] is a game based on Minecraft, and its mechanics can be understood as combining a puzzle game with a bit of visual programming. In this game, the user must create cubes with a pre-defined action that looks like a software component. With this in mind, it is possible to use the game to teach components and the importance of reusing software, forcing the user to reuse to complete the puzzle.

Lost in Space [27] was another game found that could be used to teach software components. A tittle modification in the game mechanics would be enough, the idea would be to use components to control the game ship or change the dynamics of the game and make the player have to create a component that aims to destroy the ship.

Finally, a similar puzzle game was also found to teach Java programming [45]. This game has a collection of activities divided into levels where the user needs to write code snippets, taking the character from point A to point B to avoid obstacles. This game with minor modifications could also be used to teach software components, where at each phase, the player would need to create a component to reuse throughout the other levels.

B. Derivation games

Most games are built from derivations of others with minor modifications to sprites, stories, or mechanics. The reason so many games look similar is that they use the same set of mechanics with small changes [51].

When discussing modifications and derivations of games, a well-known term in this area is mod. This term can be defined as any form of non-commercial change of a proprietary digital game [52]. Through this term, it is possible to notice that many games are built from others, for example, when entering the website moddb.com, it is possible to find more than 21000 modified projects (accessed on June 28, 2020). There are other terms to define games that were created from others. This characterization depends on the size of the modification that was made. Table VII demonstrates each of the categorizations and presents a discussion of whether such a modification is a derivation or not [52] [53].

Games are formed by different elements that combined give life to the experience lived by the player. Small changes in these elements make it possible to derive a new game that will provide new experiences. Different authors divide these elements in different ways, and in the following three ways to divide these elements will be demonstrated.

The first method of dividing game elements is MDA. This method is divided into three essential elements: mechanics, dynamics, and aesthetics. **Mechanics** can be understood as the rules and actions that can happen during the game. **Dynamics** represent the behavior that occurs as a result of actions. Finally, **aesthetics** are the emotions experienced by the player [54].

The second method of dividing the elements of games that was found was the proposal made by Jesse Schell [51]. In this method, every game consists of four essential elements: aesthetics, mechanics, history, and technology. **Aesthetics**, in this case, can be understood as what gives the appearance to the game, anything that interacts with the player, such as images and sounds. **Mechanics** is the rules and actions that can happen; **history** is the narrative of the game, the events that are happening throughout the game. Finally, **technology** is the junction point of all others; it is where the game is built [51].

The last method found was that proposed by Aki Jarvinen [55]. This method divides the elements into some groups: **mechanics** that have already been defined; **rules** that are the procedures with which the game system restricts and moderates the game; **theme** that is the subject of the game, the plot; and **information** that can be understood as what the system stores and presents to the player.

Based on these elements that make up the games, it is possible to create derivations of existing games by making small changes to parts of these elements. For example, think of a simple card game like canasta. This game has several variations that use the same technology (the cards), mechanics, aesthetics and dynamics similar to just a few changes in the rules. However, it is important to note that it is not any change in an element that generates a new game. For example, a change in technology or appearance (sprites, sounds) in the game will not create a new game. To try to make it more transparent to the reader what can be modified to derive one game from another, Table VII groups the similar elements and presents a brief discussion about each one.

Based on the idea of derivation, two games were built using the main mechanics and characteristics found in this mapping. The idea was to create games derived from those found to teach SR. The first game was a simple quiz to demonstrate the information to be taught in an SR discipline in a way that was not only through the reading of books and slides. The quiz used characteristics such as response time, demonstration of the right and wrong answers as feedback, progress bar, and difficulty levels. Thinking about derivation, this game could be easily modified and applied in other contexts different from teaching of SR. It would be enough to modify the questions in the game to satisfy the criteria of another discipline. The second game explored the derivation idea further and was inspired by the Lightbot, using a similar mechanic with some modifications. The game created was called CodeBoy and can

TABLE VI. CATEGORIZATION OF CHANGES, ADAPTED FROM [52, 53]

Name	Description	Discussion
Mutators or tweaks	Minor modification, such as changing the speed of the game. Mutators can also be additions that do not influence the game and its mechanism, having only an aesthetic effect.	It is not considered a derivation because it has only aesthetic effects, or very small modifications.
Add-ons	Provides some extensions, such as new maps, new units, and new skins, among others.	It is not considered a derivation because it is just an extension.
Mods	Can include add-ons and mutators, or rather, they also manipulate the rules system and the visual layer. Often, they try to establish a new version of the game by modifying much of it.	Alter the original game significantly, and therefore considered a derivation.
Total conversions	Manipulate the original game in so many ways that a new game is created. Still, these changes can be applied to different levels of the game.	

TABLE VII. DISCUSSION OF ELEMENTS.

Element	Discussion
Mechanic (Actions)	Modifications to the actions that the user can perform in a game will create a derivation. For example, think in a chess where the pieces have different movements.
Mechanic (Rules)	Changes in the rules that the user can execute in a game will create a derivation. For example, think of a target shooting game where the player must play blindfolded, the game will be more difficult through this new rule added.
Dynamic	Dynamics represents the behavior that occurs as a result of actions. To have a change in the dynamics of the game, you must change part of the mechanics. Therefore, an isolated change in this element would not occur.
Aesthetic	Aesthetics are the emotions experienced by the player. It will hardly be possible to make an isolated change in this element. It will be necessary to modify another element to influence it.
Aesthetic (Appearance) Information	Modifications in the appearances (sprites, sounds) of games will not create a new game. Playing Zelda without music is still the same game.
Story/Theme	Changes in stories or themes of games will result in a new game. Many developers use this approach, creating new games that only change the story. For example, there are several action games with walking and shooting mechanics but with different stories.
Technology	The change in technology does not create a new game. It can change the experiences experienced by the player. However, it remains the same game. Computer chess remains a game of chess.

be used to demonstrate the importance of SR, showing the student that there are certain moments when it is essential and often indispensable. CodeBoy used movement mechanics similar to Lightbot as jumping, rotating, and walking. This game is also integrated with the idea of the FODA (Feature-oriented domain analysis) language tree in the context of SR to permit the construction of functions [56].

The Codeboy derivation process followed three phases, namely: analysis, division, and derivation. The analysis seeks to obtain an understanding of the game to be derived, collecting information such as the objective of the game and how it works. The division aims to separate the game according to the elements of the Table VII, to understand each one separately. Finally, in the derivation phase, each element is reviewed to understand which elements will be modified or reused. Table VIII shows the division phase with each of the elements of the two games. It is worth remembering that when an element was reused, its discussion will be demonstrated only once. Figure 3 demonstrates Lightbot [29] at the top and CodeBoy [56] at the bottom. A better understanding of the games created and their evaluations can be found in Castro and Werner [56].

Figure 3. Lightbot [56] and Codeboy [29]

VI. Final Remarks

A. Threats to Validity

Through a critical analysis of the mapping, it is possible to perceive some threats that may have affected the final result of the work. The first to be highlighted is about the period in which the mapping was performed, collecting information from just five years. The second threat is the problem of interpreting the information found, which is up to the author to understand the game found and think of a way that could be applied in the teaching of SR.

TABLE VIII. LIGHTBOT VS CODEBOY.

Element	LightBot	CodeBoy
Objective	Teach programming logic through sequence of movements as an algorithm	Teach reuse through the development of software components
Mechanic (Actions)	1 - Drag and drop blocks to move the robot 2 - Create functions with a sequence of steps for character movement 3 - Turn on the lights along the way	1 - Drag and drop blocks of codes to move the boy and solve the phase algorithm 2 - Create functions with a sequence of steps for character movement 3 - Catch the gold star 4 - Open the magic chest
Mechanic (Rules)	Create the sequence of correct movements to move the robot	The basis of the game is through the construction of components. It is worth remembering that these created components can be saved and reused in later stages
Dynamic	Watch the robot move and turn on the lights along the route	After creating the algorithm, it will be validated through metrics and a score will be calculated for the player, encouraging the user to create better programs to achieve higher grades
Aesthetic	The user must think about the correct movement sequence (challenge)	The user must develop his own component (expression) based on the algorithm described in the phase, so that he can get an adequate grade to advance to the next phase (challenge)
Aesthetic (Appearance) Information	1 - Maps divided into squares where the game character moves around the squares. 2 - Group of commands to be executed positioned on the right side of the screen.	1 - Maps divided into squares where the game character moves around the squares. 2 - Group of commands to be executed positioned on the right side of the screen, within the tree.
Story/Theme	Help the robot to move from point A to point B by raising the lights along the way.	Help the character move from the start to capture the gold star and then open the treasure chest.
Technology	Mobile	Mobile e desktop

B. *Conclusion and Future Work*

For many people who are not directly linked to the software reuse area, they refer to it as just code. Due to this fact, this mapping sought to find programming teaching games that could be used to teach reuse concepts that are often abstract to many students. From this, it was possible to identify six games and six block-based programming languages. The game, and the visual programming language that were identified in more articles were LightBot [29] and Scratch [30], respectively. The main characteristics found were the use of rules, phases, difficult progression, feedback, challenges, and the use of similar tasks in sequence.

As mentioned before, software reuse is inserted in several contexts, and the most common are propagation and engineering. This work sought to identify games that were created to teach programming but could be used to explain some of the fundamentals of software reuse, thus looking at works from the first context. To better understand how these games are used as teaching methods, it is intended to perform another mapping to identify games that aim to teach software engineering, since as software reuse is inserted in the engineering and possibly similar features can be used to the teaching of the two subjects.

Although this work has found some games that could be used to teach some reuse fundamentals such as components, functions, and object orientation, none of these games were specifically designed to teach software reuse. Based on each of the games found, a possible modification or use of it for teaching SR was proposed. Based on these proposals, it was understood that games could be derived or reused for more than one context. Most games are created from others with minor modifications, this process is called derivation. In order to test this process, a three-step protocol was built and used to create a game from it.

Games can be a new method to complement the current teaching method due to its main advantages, such as increased practice and engagement through challenges, rewards, fun, and feedback. However, it is still something new that needs attention due to problems such as the complexity of the game that can affect learning, and the level of learning provided by games is still lower than current teaching methods.

As mentioned before, several block-based programming languages have been found, showing that this strategy has also been used on a large scale for teaching programming. Based on this, it is intended to create games that use this strategy to help teaching Software Reuse.

To conclude, it is expected to improve and systematize the game evolution showed in this paper to continue the derivation of games more transparently and efficiently.

REFERENCES

[1] D. Castro and C. Werner, "Mapping on the use of games for programming teaching with an emphasis on software reuse," The Sixth International Conference on Advances and Trends in Software Engineering, SOFTENG, February, 2020.

[2] J. Sametinger, Software engineering with reusable components. Springer Science & Business Media, 1997.

[3] O. Serrat, "Harnessing creativity and innovation in the workplace," in Knowledge solutions. Springer, 2017, pp. 903–910.

[4] R. J. Leach, Software Reuse: methods, models, and costs. McGraw-Hill New York, 1997.

[5] N. Niu, D. Reese, K. Xie, and C. Smith, "Reuse a" software reuse" course," in American Society for Engineering Education. American Society for Engineering Education, 2011.

[6] U. Ritterfeld, M. Cody, and P. Vorderer, Serious games: Mechanisms and effects. Routledge, 2009.

[7] H. Ludens, "A study of the play element in culture," Trans. by RFC Hull.(London, 1949), vol. 168, 1955.

[8] T. Mikkonen and A. Taivalsaari, "Software reuse in the era of opportunistic design," IEEE Software, vol. 36, no. 3, 2019, pp. 105–111.

[9] S. Combéfis, G. Beresnevičius, and V. Dagienė, "Learning programming through games and contests: overview, characterisation and discussion," Olympiads in Informatics, vol. 10, no. 1, 2016, pp. 39–60.

[10] C. Malliarakis, M. Satratzemi, and S. Xinogalos, "Educational games for teaching computer programming," in Research on e-Learning and ICT in Education. Springer, 2014, pp. 87–98.

[11] M. A. Miljanovic and J. S. Bradbury, "A review of serious games for programming," in Joint International Conference on Serious Games. Springer, 2018, pp. 204–216.

[12] I. Steinmacher, A. P. Chaves, and M. A. Gerosa, "Awareness support in distributed software development: A systematic review and mapping of the literature," Computer Supported Cooperative Work (CSCW), vol. 22, no. 2-3, 2013, pp. 113–158.

[13] B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele University, vol. 33, no. 2004, 2004, pp. 1–26.

[14] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "Characterizing interoperability in context-aware software systems," in 2016 VI Brazilian Symposium on Computing Systems Engineering (SBESC). IEEE, 2016, pp. 203–208.

[15] S. Matalonga, F. Rodrigues, and G. H. Travassos, "Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review," Journal of Systems and Software, vol. 131, 2017, pp. 1–21.

[16] M. Petticrew and H. Roberts, Systematic reviews in the social sciences: A practical guide. John Wiley & Sons, 2008.

[17] S. Jiang, H. Zhang, C. Gao, D. Shao, and G. Rong, "Process simulation for software engineering education," in Proceedings of the 2015 International Conference on Software and System Process. ACM, 2015, pp. 147–156.

[18] A. Calderón, M. Trinidad, M. Ruiz, and R. V. O'Connor, "Teaching software processes and standards: A review of serious games approaches," in International Conference on Software Process Improvement and Capability Determination. Springer, 2018, pp. 154–166.

[19] S. Ramírez-Rosales, S. Vázquez-Reyes, J. L. Villa-Cisneros, and M. De León-Sigg, "A serious game to promote object oriented programming and software engineering basic concepts learning," in 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT). IEEE, 2016, pp. 97–103.

[20] K. Salen, K. S. Tekinbaş, and E. Zimmerman, Rules of play: Game design fundamentals. MIT press, 2004.

[21] C. Caulfield, J. C. Xia, D. Veal, and S. Maj, "A systematic survey of games used for software engineering education," Modern Applied Science, vol. 5, no. 6, 2011, pp. 28–43.

[22] T. Jordine, Y. Liang, and E. Ihler, "A mobile-device based serious gaming approach for teaching and learning java programming," in 2014 IEEE Frontiers in Education Conference (FIE) Proceedings. IEEE, 2014, pp. 1–5.

[23] E. Lee, V. Shan, B. Beth, and C. Lin, "A structured approach to teaching recursion using cargo-bot," in Proceedings of the tenth annual conference on International computing education research. ACM, 2014, pp. 59–66.

[24] R. O. Chaves, C. G. von Wangenheim, J. C. C. Furtado, S. R. B. Oliveira, A. Santos, and E. L. Favero, "Experimental evaluation of a serious game for teaching software process modeling," ieee Transactions on Education, vol. 58, no. 4, 2015, pp. 289–296.

[25] F. Agalbato, "Design and implementation of robo3: an applied game for teaching introductory programming," Scuola di Ingegneria Industriale e dell'Informazione, 2017.

[26] R. Atal and A. Sureka, "Anukarna: A software engineering simulation game for teaching practical decision making in peer code review." in QuASoQ/WAWSE/CMCE@ APSEC, 2015, pp. 63–70.

[27] Á. Serrano-Laguna, J. Torrente, B. M. Iglesias, and B. Fernández-Manjón, "Building a scalable game engine to teach computer science languages," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 10, no. 4, 2015, pp. 253–261.

[28] M. M. Burnett, "Visual programming," Wiley Encyclopedia of Electrical and Electronics Engineering, 2001.

[29] E. V. Duarte and J. L. Pearce, "A cross-cultural review of lightbot for introducing functions and code reuse," Journal of Computing Sciences in Colleges, vol. 33, no. 2, 2017, pp. 100–105.

[30] D. Topalli and N. E. Cagiltay, "Improving programming skills in engineering education through problem-based game projects with scratch," Computers & Education, vol. 120, 2018, pp. 64–74.

[31] R. Law, "Introducing novice programmers to functions and recursion using computer games," in European Conference on Games Based Learning. Academic Conferences International Limited, 2018, pp. 325–334.

[32] C.-K. Chang, "Effects of using alice and scratch in an introductory programming course for corrective instruction," Journal of Educational Computing Research, vol. 51, no. 2, 2014, pp. 185–204.

[33] N. Pellas and S. Vosinakis, "Learning to think and practice computationally via a 3d simulation game," in Interactive Mobile Communication, Technologies and Learning. Springer, 2017, pp. 550–562.

[34] M. Aedo Lopez, E. Vidal Duarte, E. Castro Gutierrez, and A. Paz Valderrama, "Teaching abstraction, function and reuse in the first class of cs1: A lightbot experience," in Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education. ACM, 2016, pp. 256–257.

[35] N. Boldbaatar and E. Şendurur, "Developing educational 3d games with starlogo: The role of backwards fading

in the transfer of programming experience," Journal of Educational Computing Research, vol. 57, no. 6, 2019, pp. 1468–1494.

[36] J. Vincur, M. Konopka, J. Tvarozek, M. Hoang, and P. Navrat, "Cubely: Virtual reality block-based programming environment," in Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology, ser. VRST '17. New York, NY, USA: ACM, 2017, pp. 84:1–84:2. [Online]. Available: http://doi.acm.org/10.1145/3139131.3141785

[37] M. Marimuthu and P. Govender, "Perceptions of scratch programming among secondary school students in kwazulu-natal, south africa," African Journal of Information and Communication, vol. 21, 2018, pp. 51–80.

[38] N. Pellas, "The development of a virtual learning platform for teaching concurrent programming languages in the secondary education: The use of open sim and scratch4os," Journal of e-Learning and Knowledge Society, vol. 10, no. 1, 2014, pp. 129–143.

[39] H. Pötter, M. Schots, L. Duboc, and V. Werneck, "Inspectorx: A game for software inspection training and learning," in 2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T). IEEE, 2014, pp. 55–64.

[40] N. Tabet, H. Gedawy, H. Alshikhabobakr, and S. Razak, "From alice to python. introducing text-based programming in middle schools," in Proceedings of the 2016 ACM Conference on innovation and Technology in Computer Science Education, 2016, pp. 124–129.

[41] F. Kalelioglu and Y. Gülbahar, "The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective." Informatics in Education, vol. 13, no. 1, 2014, pp. 33–50.

[42] L. A. Vaca-Cárdenas, F. Bertacchini, A. Tavernise, L. Gabriele, A. Valenti, D. E. Olmedo, P. Pantano, and E. Bilotta, "Coding with scratch: The design of an educational setting for elementary pre-service teachers," in 2015 International Conference on Interactive Collaborative Learning (ICL). IEEE, 2015, pp. 1171–1177.

[43] S. Yamashita, M. Tsunoda, and T. Yokogawa, "Visual programming language for model checkers based on google blockly," in International Conference on Product-Focused Software Process Improvement. Springer, 2017, pp. 597–601.

[44] J.-M. Sáez-López, M. Román-González, and E. Vázquez-Cano, "Visual programming languages integrated across the curriculum in elementary school: A two year case study using "scratch" in five schools," Computers & Education, vol. 97, 2016, pp. 129–141.

[45] A. Sierra, T. Ariza, F. Fernández-Jiménez, J. Muñoz-Calle, A. Molina, and Á. Martín-Rodríguez, "Educational resource based on games for the reinforcement of engineering learning programming in mobile devices," in 2016 Technologies Applied to Electronics Teaching (TAEE). IEEE, 2016, pp. 1–6.

[46] D. Valencia, A. Vizcaino, L. Garcia-Mundo, M. Piattini, and J. P. Soto, "Gsdgame: A serious game for the acquisition of the competencies needed in gsd," in 2016 IEEE 11th International Conference on Global Software Engineering Workshops (ICGSEW). IEEE, 2016, pp. 19–24.

[47] D. Bau, "Droplet, a blocks-based editor for text code," Journal of Computing Sciences in Colleges, vol. 30, no. 6, 2015, pp. 138–144.

[48] D. A. Bau, "Integrating droplet into applab—improving the usability of a blocks-based text editor," in 2015 IEEE Blocks and Beyond Workshop (Blocks and Beyond). IEEE, 2015, pp. 55–57.

[49] H. Chandrashekar, A. G. Kiran, B. Uma, and P. Sunita, "Introducing programming using "scratch" and "greenfoot"," Journal of Engineering Education Transformations, 2018.

[50] R. d. A. Mauricio, L. Veado, R. T. Moreira, E. Figueiredo, and H. Costa, "A systematic mapping study on game-related methods for software engineering education," Information and software technology, vol. 95, 2018, pp. 201–218.

[51] J. Schell, The Art of Game Design: A book of lenses. CRC press, 2008.

[52] A. Unger, "Modding as part of game culture," in Computer Games and New Media Cultures. Springer, 2012, pp. 509–523.

[53] O. Sotamaa, "When the game is not enough: Motivations and practices among computer game modding culture," Games and Culture, vol. 5, no. 3, 2010, pp. 239–255.

[54] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in Proceedings of the AAAI Workshop on Challenges in Game AI, vol. 4, no. 1, 2004.

[55] A. Järvinen, "Introducing applied ludology: Hands-on methods for game studies." in DiGRA Conference. Citeseer, 2007.

[56] D. Castro and C. Werner, "Use of games as a strategy for teaching software reuse," SBGAMES, 2020 (in Portuguese).

Microservice-Enabled Simulation Platform for Industry-4.0 Dynamic Systems: A Computational Fluid Dynamics Case Study for Underground Coalmine Ventilation Networks

Alexey Cheptsov
High Performance Computing Center Stuttgart,
University of Stuttgart
Stuttgart, Germany
e-mail: cheptsov@hlrs.de

Oleg Beljaev
Donetsk National Technical University,
Computer Science Department
Pokrowsk, Ukraine
e-mail: oleg69@ukr.net

Abstract—**Industry 4.0 is a state-of-the-art methodology of complex industrial systems development, which aims to improve the industrial processes by applying the digitalization and computer processing to the technology-level of production objects. The ability to process streaming data from numerous digital sensors, as enabled by Industry 4.0, endorses a potential for creation of many innovative applications. One of the major classes of industrial applications that are enabled by Industry 4.0 is the simulation – the technology that is frequently used for the "offline" optimization of technological processes. In the context of Industry 4.0, the simulation will be benefitting from the "online" integration with the digitalized industrial infrastructure, e.g., for the realization of real-time support scenarios. However, the existing simulation approaches and tools are not able to support highly dynamic, hierarchically organized industrial systems due to a monolithic design of those tools, motivated by the other essential requirements, such as efficiency, user- and developer-friendliness, etc. This article proposes an approach that is based on microservices – the functionally-decoupled, interconnected composition blocks of a hierarchically organized, modular simulation application, that implements a specific part of the simulation logic for the targeted physical phenomena. The use of the microservice-based approach is demonstrated on the implementation of a simulation framework for underground coalmine ventilation networks – complex technological objects that impose challenging tasks, such as conduction of Computational Fluid Dynamics studies.**

Keywords-microservices; dynamic systems; real-time simulation; computational fluid dynamics; cyber-physical automation; ventilation network.

I. INTRODUCTION

Modelling and Simulation are essential technologies for design and support of complex industrial and technological objects, which can be methodologically considered as **Dynamic Systems (DS)**. DS form a special class of simulation applications that are based on the (mathematical) models that are built according to the functional and structural decomposition approaches. The essence of those approaches lies in the application of a numerical decomposition method (e.g., domain decomposition, functional decomposition, structural decomposition, etc.) in order to i) break a complex system down into the elementary sub-systems (sub-models), which can be represented by means of the standard mathematical techniques (e.g., systems of ordinary and/or differential equations), as well as ii) define a methodology to combine those sub-models to solve the original big problem.

A typical DS example is a **Ventilation Network (VN)** of an underground coalmine, as discussed in our earlier publication [1] and also taken as the basis case study in this article (see a further description in Section II). VNs are challenging objects in terms of design, operational support, and control – they expose several safety-critical tasks, which cannot be solved in the practice without an extensive support, provided by the simulation tools. One of such tasks is the analysis of the **aerodynamics**, i.e., of the distribution of the air and gases in underground mining areas – the major factor of the coalmine production safety. The classic simulation approach for such tasks is served by the **Computational Fluid Dynamics (CFD)** technique – a numerical computation method based on complex models.

The quality of the CFD models prediction (the model data accuracy) can be considerably improved by incorporation of the data that are measured at the object (i.e., the physical measurements data) into the simulation – the strategy that is often referred as a part of the "cyber-physical system" or "internet-of-things" concept (see a survey of Bordel Sanchez and Alcarria [2]). Nowadays, these concepts are considered as essentials of the **Industry 4.0** methodology, aiming to maximize the digitalization and automation of industrial systems (see overview of Wollschlaeger et al. [3]). For this purpose, the modern industrial technological objects are being increasingly equipped with "smart sensors" – light-weight measurement devices with rich communication interfaces (like Ethernet) that allow retrieval of physical data, e.g., in form of the real-time data streams. The new-generation smart sensors are characterized by a wide portability and low power consumption and hence can be installed in a very broad spectrum of industrial technological objects, also including the production areas of coalmines and their VNs. The data that are obtained from the sensors can be used not only to "calibrate" the simulation experiments results, but also to support the models in the situations when they cannot be reliably applied to some specific parts of the modelled physical problem or phenomena, e.g., those that include unpredictable processes, such as the gas emission from dynamic sources of VNs, such as the goaf – a cavity in the production areas of the underground coalmining works. In particular, the published work of Stewart et al. [4] has shown that the value of VN models can be considerably increased for the use in real, productional cases by incorporating the sensors data. However, integration of sensor data into the general-purpose simulation packages is not easy, in particular, due to the monolithic design of the former.

This paper evaluates the major problems with which the simulation experts might face during the development of sensor-aware simulation application scenarios by using wide-spread programming models and tools. As an efficient and, at the same time, developer-friendly alternative to the traditional approaches, a decentralized, microservice-oriented simulation framework *MS-Sim* is proposed and described.

The remainder of the paper is organized as follows: Section II gives an overview of VNs and modelling techniques for their analysis. Section III introduces and discusses the methodology of a real-time simulation of VNs. Section IV is dedicated to the microservice architecture for development of simulation applications. Section V presents the evaluation use cases and discusses the validation and benchmarking results. Section VI concludes the paper and discusses the main outcomes.

II. Overview of Ventilation Networks as Objects of Modelling and Simulation

Despite of the revolutionary development in the domain of renewable and regenerative energy technologies, as presented in the last decade, (black) coal remains one of the most important sources of energy for industrial and urban objects world-wide, along with the gas, oil and other fossil fuels. The total discovered reserves of *1 Tera-ton* along with the actual annual production of *5,5 Giga-ton* (according to the International Energy Program [5]) makes the coal to one of the most essential energy sources of the world's industry, especially in the regions of East Europe, Asia, and Australia. At the same time, the coal industry is one of the most unsafe industrial sectors in terms of the work safety – the specific of the underground production (carried out at the depths of up to many kilometers under the surface), such as a limited ability of the natural air circulation through the underground mining fields or a frequent location of dynamic sources of hazardous gases, cause in the practice a considerable percentage of victims among the work labour due to technological accidents, such as the explosions of methane and oxygen mixtures.

Ventilation is the most important technology that is used for the safety ensuring in underground production areas of coalmines – it aims to provide mining fields with the fresh air conducted from the surface (in the amount that is necessary for carrying out the mining processes by the underground workers), as well as to drain the hazardous gases (mainly, CH_4 – methane) that are emitted during the mining and coal transporting from the underground production areas. The real VNs have to cover a widely distributed network of diverse interconnected underground airways (see Fig. 1) that span over many mining sections (e.g., longwalls) with diverse elements with the total length of up to many thousands of kilometers, as shown in Fig. 2. The ventilation is enforced by the *ventilation fans* (along with smaller local booster fans), mounted on the surface, that ensure an uninterrupted air circulation through all the elements of the VN sections: *main gates* (leading the fresh air from the surface), *faces* (areas of coalmining), *tail gates* (offtakes of air-gas mixtures), etc.

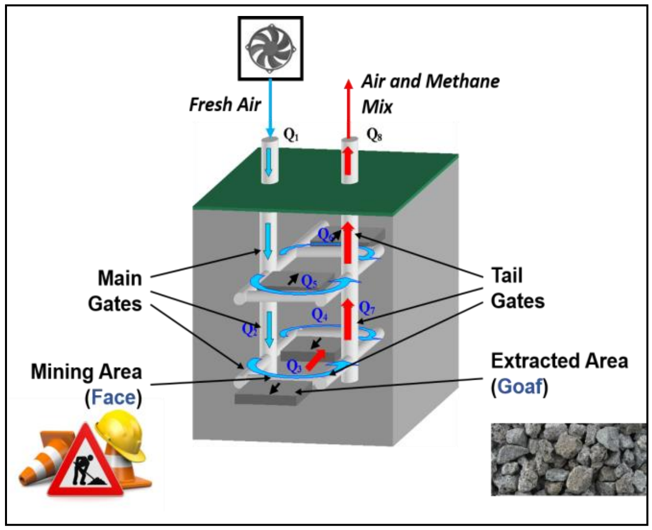


Figure 1. General structure of mine ventilation section.

The engineering analysis of the VN ventilation aims to solve a multi-level (including the airway, section, or network levels) regulation problem in order to determine operation modes of the major and booster fans (i.e., the depression values), values of the local regulators (e.g., the positions of *sluice valves*), etc., in order to saturate every VN airway with the airflow-rates required for the safe coalmining production. In practice, this is frequently performed with the help of static models, having the following general form:

$$P = RQ^2, \qquad (1)$$

where P – the air-pressure across the airway, Q – the air-flow rate, R – the air-flow resistance.

The air-flow model in the general form (1), applied to each and every VN element, results in the equation system that allows calculation of static ventilation properties for the whole VN. However, dealing with more complex tasks like the ventilation on demand (ensuring optimal energy consumption in constantly changing VN configurations due to gradual progress of the coalmining process), or solving any task that requires analysis of the gas emission from the coal-bed implies the analysis of dynamic behavior and makes the static model (1) inapplicable. Instead, the dynamic models are required at this place.

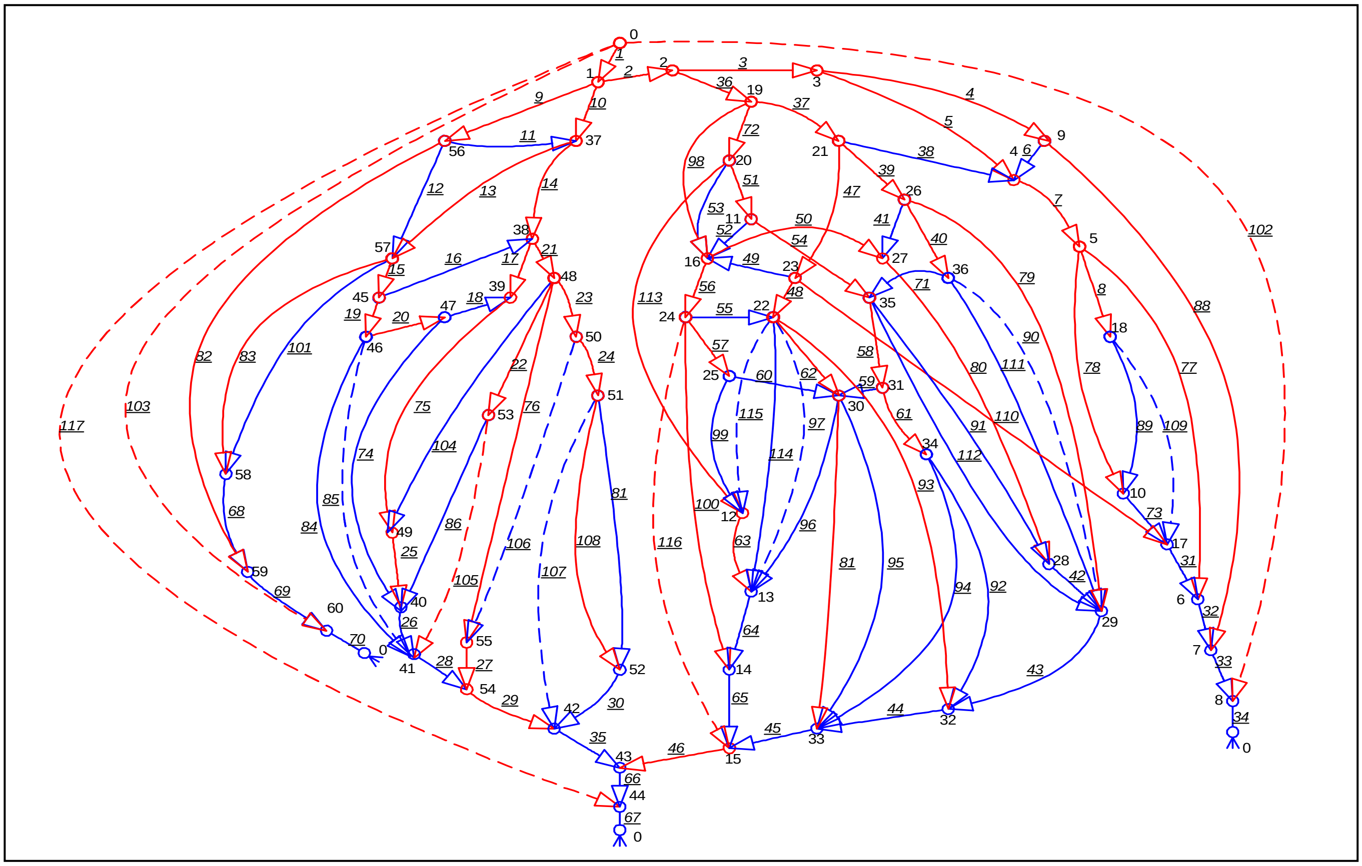


Figure 2. Illustration of real-complexity ventilation network with 117 branches and 61 connection nodes – coalmine South-Donbass-3, located in Ukraine. *The coloring of airways is used here just for better readibility purpose*.

The latter analysis task – gas-dynamics – is of a special importance for the coalmine safety, as most of the emergency situations in the coalmining industry happen due to combustion of the gases preserved in the coal veins [6]. Once past the *face*, the intake air that travels up the *main gate* and leaves the section through *the tail gate* is no longer fresh, but contaminated by mine gases, such as methane. Methane may be emitted from diverse dynamic sources in the section. The major source is, however, the *goaf* – a cavity between the *main* and the *tail gates*, which preserves the sealed methane that is carried out by the air-flows bleeding through it (parallel to the *face*, as depicted in Fig. 3). The concentration of the methane in the mixture with oxygen carried away by the *main gate*'s bleeding flows through the *goaf* may relatively quickly reach the upper limit (of approx. 5%), which can cause spontaneous combustions of gases and coal in the *goaf*. Therefore, the gas-dynamics models are of a special importance for the coalmine safety, for the reason that they allow a prediction of potentially unsafe situations.

The gas-dynamics analysis relies on the models that consider dynamic aspects of the unsteady, multiphase (air- and gas-) flows, which are a way more complex than the static model (1). The general approach to the analysis of the dynamic problems involving fluids is the use of **Navier-Stokes (NS)** equations for a compressible flow (e.g., see a detailed explanation in Chorin [7]), covering different aspects of the fluids' macroscopic properties, such as the free-streaming and interaction with complex-geometry surfaces. Since the NS equations are non-linear and have no analytical solution, numerical techniques have to be applied. The common practical way used in the numerical CFD simulation is to apply an approximation and decomposition in order to simplify the original mathematical model whilst keeping the most essential fluid properties (e.g., flow rates, viscosity, pressures, etc.) sufficiently represented. A widely used VN aerodynamic model is a 1-D NS-approximation proposed by Svjatnjy [8], in the form of the equation system (2), consisting of a transport (impulse) and a continuity equation.

$$\begin{cases} -\dfrac{\partial P}{\partial \xi} = -\dfrac{2\rho}{F^2} Q \dfrac{\partial Q}{\partial \xi} + \dfrac{\rho}{F}\dfrac{\partial Q}{\partial t} + rQ^2 + r'(t)Q^2 \\ -\dfrac{\partial P}{\partial t} = \dfrac{\rho a^2}{F}\dfrac{\partial Q}{\partial \xi} - \dfrac{\rho a^2}{F} q \end{cases}, \quad (2)$$

where t – the time, ξ – the spatial coordinate, and other values – diverse aerodynamic properties and coefficients.

The NS-based equations can also describe the state and transport properties of gas fluids. One of the “classic” models that describe a methane emission from the *goaf*'s filtration space was proposed by Feldmann [9]:

$$\int_V m \frac{\partial(\rho C)}{\partial t} dV = G_0 - G, \qquad (3)$$

where G – the gas-mass flow that is carried out by filtration air-flow from the goaf, G_0 – the gas-mass flow in the filtration volume of the goaf, C – the gas concentration in the filtration flow through the goaf, m – the coefficient of the goaf porosity, as well as other aerodynamic properties.

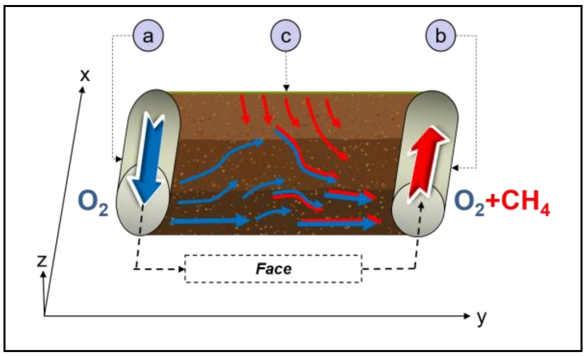


Figure 3. Methane emission schema in a U-type (longwall) section ventilation: a) main gate, b) tail gate, c) goaf.

The Feldmann's model (3) of the methane emission from the *goaf* can be approximated (in order to simplify the further numerical solution) by a 1-D representation, e.g., in the following form, as elaborated by Svjatnjy [8]:

$$\frac{mV}{q}\frac{dQ_m}{dt} + Q_m = Q_{m0} + \frac{mV}{q^2} Q_m \frac{dq}{dt}, \qquad (4)$$

where Qm – the methane-flow rate and q – the air-flow rate in the filtration space of the goaf.

The methane transportation by the flow in the section's *tail gate*, surrounded by the *goaf*, as well as in the further VN airways along the air-flow contour direction can be described, for example, by the following model built according to the Stewart's discrete cells approach [4]:

$$Q^i_{mt} = \sum_{j=1}^{N} Q^j_{mt}, \qquad (5)$$

where Q^i_{mt} – the rate of the methane-flow that is transported by the i-discretization element of the airway, N – the number of discretization elements that act as sources of the methane transported with the incoming air-flow.

The overall dynamic model of the complete VN can be built up from the models of each of its elements, i.e.:

- airways with only air-flow (e.g., the *main gates* leading fresh air to the section), such as represented by model (2),
- airways that contain active methane sources (e.g., the *goaf*), such as represented by models (2) and (4),
- airways that are transporting both air- and methane-flows from the adjoining airways (e.g., the *tail gates* carrying out the used air and methane from the section), such as represented by model (2) and (5),

as well as by the connective equations that describe the boundary conditions between the elements, as defined at the phase of discretization. Discretization can be performed by means of several methods, such as the Finite Differences (FDM), Finite Volume (FVM), Discontinuous Galerkin (DM), etc., see more details at [10]. Fig. 4 shows an example of a 1-D FDM discretization for a simple mining section containing a *main gate* (air-flow Q_{MG}), a *face* (air-flow Q_F), a *goaf* (air-flow Q_G and emitted gas-flow Qm_G), and a *tail gate* (air-flow Q_{TG} and gas-flow Qmt_{TG}) – see more details on FDM discretization for NS-based equations at [11].

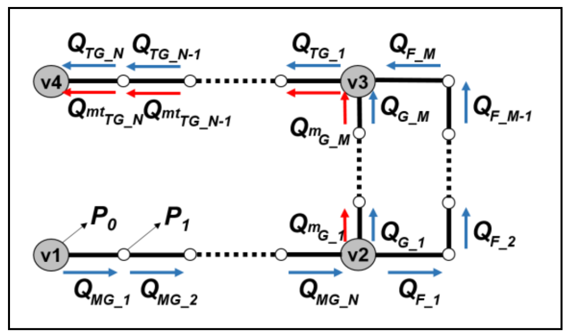


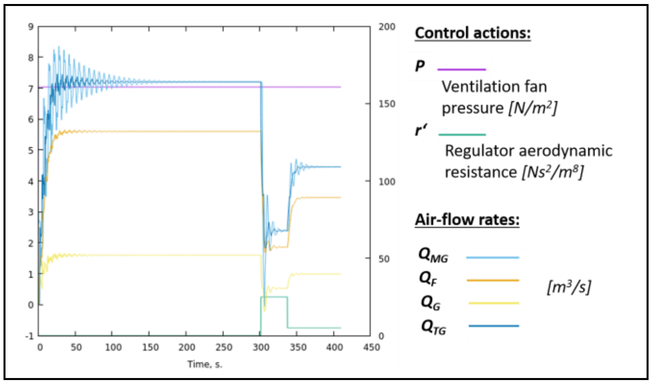
Figure 4. 1-D discretization schema of a mining section with a main gate (MG), face (F), goaf (G), and tail gate (TG).

The resulting 1-D model of a coalmining section can thus be shaped into the following form:

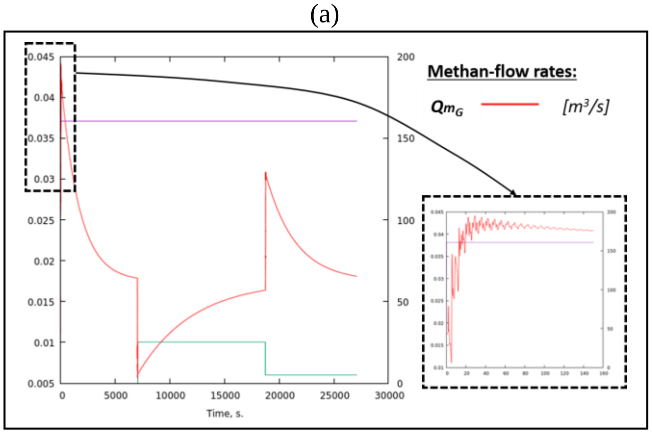
$$\begin{cases} \dfrac{\partial Q_{MGj}}{\partial t} = -\dfrac{S_j}{\rho}\dfrac{(P_i - P_{i-1})}{\Delta\psi} - rQ^2_{MGj} - r'Q^2_{MGj} \\ \dfrac{\partial Q_{Fj}}{\partial t} = -\dfrac{S_j}{\rho}\dfrac{(P_i - P_{i-1})}{\Delta\psi} - rQ^2_{Fj} - r'Q^2_{Fj} \\ \dfrac{\partial Q_{Gj}}{\partial t} = -\dfrac{S_j}{\rho}\dfrac{(P_i - P_{i-1})}{\Delta\psi} - rQ^2_{Gj} - r'Q^2_{Gj} \\ \dfrac{\partial Q_{TGj}}{\partial t} = -\dfrac{S_j}{\rho}\dfrac{(P_i - P_{i-1})}{\Delta\psi} - rQ^2_{TGj} - r'Q^2_{TGj} \\ \dfrac{mV}{Q_{Gj}}\dfrac{dQm_{Gj}}{dt} + Qm_{Gj} = Qm_{0Gj} + \\ \qquad \dfrac{mV}{Q^2_{Gj}} Qm_{Gj} \cdot \left(-\dfrac{S_j}{\rho}\dfrac{(P_i - P_{i-1})}{\Delta\psi} - rQ^2_{Gj} \right) \\ \forall_{j=1}, Qmt_{TGj} = Qm_{Gm} \\ \forall_{j=2..N}, Qmt_{TGj} = Qmt_{TG(j-1)} \\ \dfrac{\partial P_i}{\partial t} = -\dfrac{pa^2}{S_i}\dfrac{(Q_{j+1} - Q_j)}{\Delta\psi} \end{cases}, \qquad (6)$$

where $\Delta\psi$ – the length of a spatial discretization element along the air-flow contour direction.

Application of a numerical solution method (such as Euler, Runge-Kutta, etc. – see more at Feldmann's survey [12]) leads to a resolution of the equation, in its general form (6), with the aim to obtain the evolution of the dynamic processes in time. Fig. 5 shows an example of the aerodynamics processes development by applying different control actions in a test section, controlled by a *ventilation fan* and a *local regulator*, installed in the *tail gate*.



(a)



(b)

Figure 5. Example of aerodynamics analysis by means of models: (a) air-flow in langwall, (b) methane-flow in mining section's goaf.

III. State of the Art in Simulation Technologies for Ventilation Networks Analysis

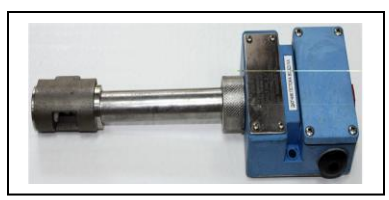
Implementation of the VN aerodynamics equations, such as needed for the ventilation control, happening on the basis of equations in the general form (6), can be performed in several ways. The simplest, in terms of developments efforts, is the usage of a wide-spread simulation package, such as *ANSYS-CFX* or *OpenFOAM*. Those packages provide a rich library of diverse solvers that can be applied on the target model equations. The other alternatives include:

- development of the own simulation platform (including discretization approaches, numerical solvers, visualization software, etc.) with more problem-specific customization options, however, requiring much more development and support actions, and
- use of the mixed (heterogeneous) environments with the customization of some specific functions of the standard simulation package.

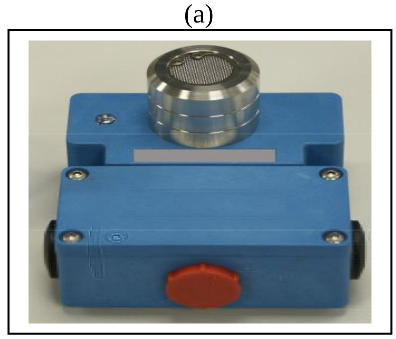
However, regardless of the chosen implementation strategy and technology, the simulation applications have to deal with one critical problem – the model data actuality. Since the VNs are highly dynamic objects, meaning their parameters are continuously changing, at the latest after every production cycle (e.g., after an extension of the section, redirection of the air-flow contour), it is important to keep the models being constantly updated, in order to ensure that they can still be applicable on the production processes in their actual state and deliver trustful results.

In the reality, it has always been very difficult, if not even impossible, to refresh the data frequently. The so-called "ventilation surveys" – physical measurements of the aerodynamics parameter in the VN – could only be conducted for every 2 to 4 weeks, for the reasons of technological complexity. However, the parameter values between the last and the new measurements might change considerably, most notably – for the gas-dynamics. Therefore, the dynamic models have not found a "critical mass" application in the coal industry up to nowadays.

However, the things have been changing in the last years rapidly – the modern coalmines rely extensively on the **Cyber-Physical Automation (CPA)**, which offers, among other advantages, an advanced data acquisition and monitoring technology, supported by the Industrial Ethernet technologies like *PROFINET*, *Modbus*, etc., see some examples by Lee at [13] or Kay at [14]. One of the CPA systems that has successfully been taken in production is *UTAS*, widely utilized at the East-European coalmining industry space, for details look at Ignatovych [15]. The sensor network of UTAS provides both air- and methane-flow detectors, see in Fig. 6.



(a)



(b)

Figure 6. Example of UTAS smart sensors: (a) Methane detector TX3263D, (b) Air-flow rate measurement station TX1322D.

The key advantage of the CPA organization for the implementation of ventilation control is an immediate (i.e., in the real-time) availability of sensors data, which might be used for the fine-grain control of technological processes. Moreover, many CPA systems serves at the same time as a decentralized computation platform for "in situ" sensor data processing, i.e., when the sensor data are processed locally on the available digital resources – the so-called "sensor processors". The interconnection of all sensor processors allows the realization of global control operations, i.e., for the whole VN. On the other hand, the distributed, heterogeneous CPA infrastructure, which is preferable to be used for conduction of CFD simulation studies straightforwardly in the industrial environment, poses a major problem for the development of the simulation software, which cannot rely on the general-purpose development environments of the simulation software, such as the above mentioned *ANSYS-CFX* or *OpenFOAM*, designed for densely-organized, massively-parallel infrastructures – the High Performance Computers (HPC) or Cloud. Even for the custom simulation frameworks that partially rely on standard packages, there are still numerous adaptations and optimizations necessary, which are very difficult to implement, e.g., as the interaction between the sensor and modelling parts of the simulation workflow.

Development of the simulation software "from scratch" would allow to overcome the general compatibility problems of the standard simulation packages. However, it is a complex and non-trivial process. Moreover, the following CPA-specific infrastructure factors have to be considered:

- Heterogeneity of the CPA distributed infrastructure – many sensor devices are provided on the basis of a host system, whose architecture might differ from the typical HPC, Cluster, or Cloud environment but still requires a seamless integration within the distributed application workflows. However, the standard parallelization approaches require a uniform infrastructure with the compute nodes of the same hardware architecture and performance class.
- Limited flexibility of the mainstream parallelization approaches to support distributed application scenarios – many parallel applications rely on the Single Instruction Multiple Data (SIMD) technique, which mainly targets densely built compute systems like HPC. However, the applications that are running on the truly distributed infrastructures (HPC + Cloud + remote embedded systems) have to be developed according to the MIMD approach, in order to allow different functionalities to be executed on different types of systems.
- Composition complexity of the CFD models – models are often organized according to the applied mathematical decomposition technique to the initial physical problem, which have to consider different hierarchical levels with complex informational and control dependencies between the composition blocks of the models (see Fig. 7). The hierarchical physical connections between the objects have to be reflected in their models, accordingly.

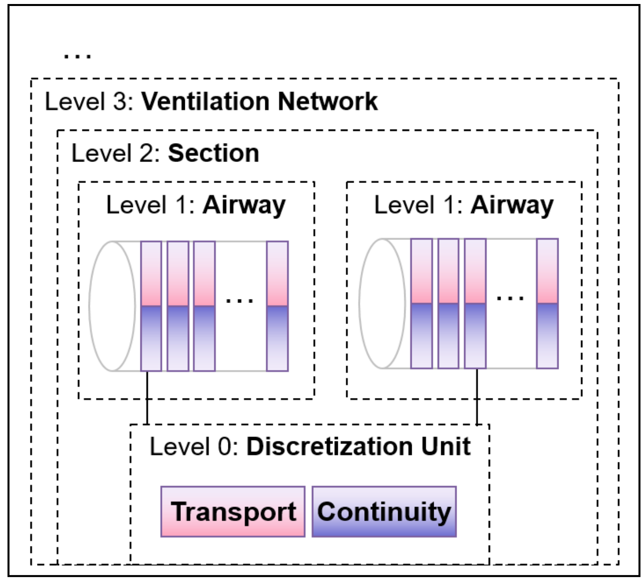


Figure 7. Hierarchical structure of composition blocks of ventilation network CFD models.

The simulation software needs to be developed in a modular way, where the modules would correspond to a part of the modelled physical system (according to the applied decomposition strategy), with the possibility of reuse of the topologically-assembled modules to implement the functionality of the higher-level models, i.e., in a bottom-up development approach. Therefore, novel development approaches are required for the implementation of the portable, horizontally and vertically scalable, as well as energy-efficient simulation software.

IV. Microservice Architecture for Development of Simulation Applications

In the last decade, **Service-Oriented Architectures (SOA)** have been successfully established in software development area as a more efficient (in terms of the required development efforts but also in terms of performance) alternative to the monolithic software design. The key concepts of SOA, such as the component-based application design, remote access of components via standardized communication protocols, etc., make the service-oriented development concept attractive for use in the simulation applications as well. However, due to the required peak performance on the highly optimized infrastructures, such as HPC, which could not be guaranteed by the standard SOA technologies, like the web services, the SOA concept has not found a wide application in the simulation applications domain. However, with i) the proliferation of the less strict, in terms of performance requirements, Cloud technology into the simulation domain, ii) appearance of heterogeneous Cloud-HPC testbeds, as well as iii) optimization of the SOA communication technology stack, the interest of the community in using the SOA technology started to grow again. With respect to the requirements discussed in previous Section III, it is proposed to use a **Microservice** (**MS**) architectures for the realization of the VN simulation framework (see Fig. 8).

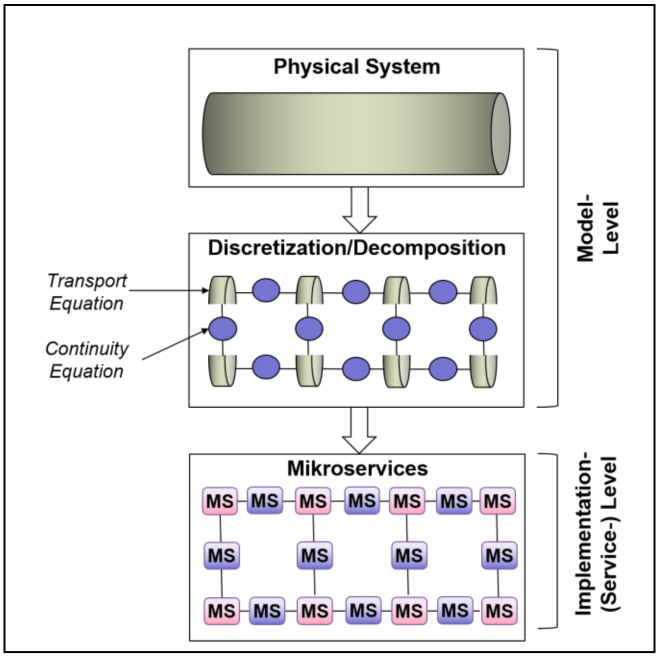


Figure 8. Composition of services for realisation of simulation application.

In this approach, a simulation application is arranged as a collection of independent, isolated, portable software modules (i.e., services), each of those implementing a composition block of the complex system, composed according to the functional, spatial, or any other decomposition. Each MS follows in its implementation a locality principle, i.e., bearing the responsibility for the assigned part of the complex system. In order to reflect the physical or informational connections of the real object, MS can be interconnected by means of a common data and/or control flow, which can be implemented with the help of a light-weight communication library, such as the technology-agnostic *HTTP*.

In order to evaluate the usability of the MS concept to carry out the VN simulation studies, as previously described in Section II, a reference framework *MS-Sim* was developed, available at the open-source platform Github [16]. The *MS-Sim* key concepts that are relevant to the simulation applications developments are as follows:

- Microservice model – specifies the development scenario for independent MS. The collection of different MS can be combined in a "market place" – library of domain-related services, which can be integrated into a common (simulation) application.
- Application model – defines the implementation strategy for MS-based (simulation) applications.
- Communication model – provides a realization of data exchange (e.g., control and data flow) between the loosely-coupled over the distributed infrastructure resources MS, as well as data storage.
- Hierarchy model – defines the implementation of hierarchically-dependent MS at the different granularity levels.

The *MS-Sim microservice model* provides a specification for every individual MS-module. According to it, MS are asynchronous and independent software processes, which are identified by a unique *Identifikator*. The MS can be data-synchronized with the other services by means of *ports* – special buffers in which the service can put the data that need to be communicated to the other services, or also obtain the incoming data (see Fig. 9). *MS-Sim* provides API to read/write data from/to buffers as well as to flush (by the sender) and synchronize (by the receiver) the buffer content.

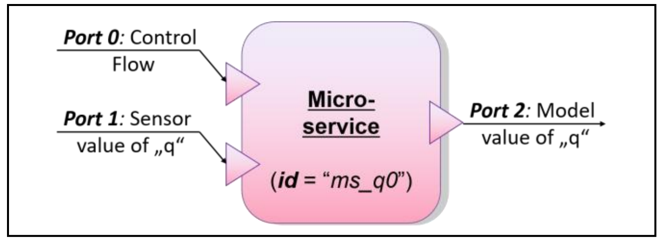


Figure 9. High-level microservice archtiecture.

At the time of service development, the connections between the MS, i.e., sources and destinations of data exchanged by means of *ports*, as well as the size of the ports' buffer, are irrelevant and can be established in the phase of creating a microservice application, i.e., by the MS-Sim *application model*. The application model specifies all the MS that are included into the application as well as connections between their corresponding ports, see Fig. 10. The MS are normally implemented in accordance to a pro-active logic, i.e., they are steered by a dedicated "master" service by means of a command flow – messages coded corresponding to a user-defined protocol, acknowledged across all MS, see example in Fig. 11. The application model distinguishes 2 categories of ports: normal buffers and *proxies*. The latter are used for internal communication with the dependent MS (e.g., between the master and the workers) and are not visible to the other application components interfacing this MS.

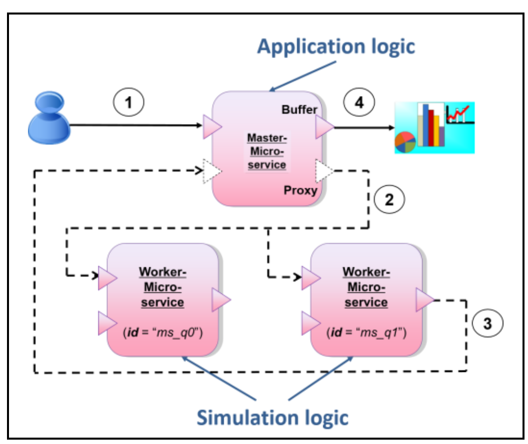


Figure 10. High-level microservice archtiecture.

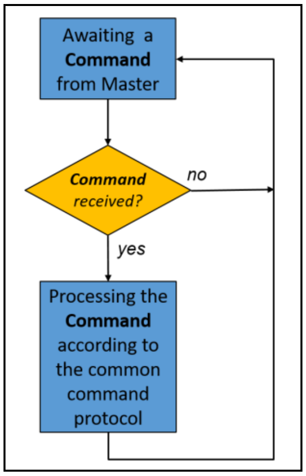


Figure 11. Implementation of command flow by worker microservices.

The typical application workflow contains instructions to initiate some important for the simulation logic operations, such as the initialization of model's parameters (e.g., by the value obtained from the sensor), initiation of the simulation, control of the simulation experiment convergence, output of results, etc.

The data exchange between the master and the workers (e.g., for the propagation of the command flow or gathering results), as well as among the workers at the time of performing the simulation (e.g., for the implementation of the boundary conditions exchange), happens asynchronously and supported at the runtime of the (simulation) application execution by the MS-Sim *communication model*. In the communication model, the connection pairs (i.e., *communication links*) between the sender and the recipient MS are specified, as shown in Fig. 12. The information about the amount (number of entities) of data transmitted between the sender's and the recipient's ports may be added to the communication link as well (or assumed as a single data value, otherwise). At the physical network level, the data transport is performed by a special communication library, supplied to the microservice execution framework.

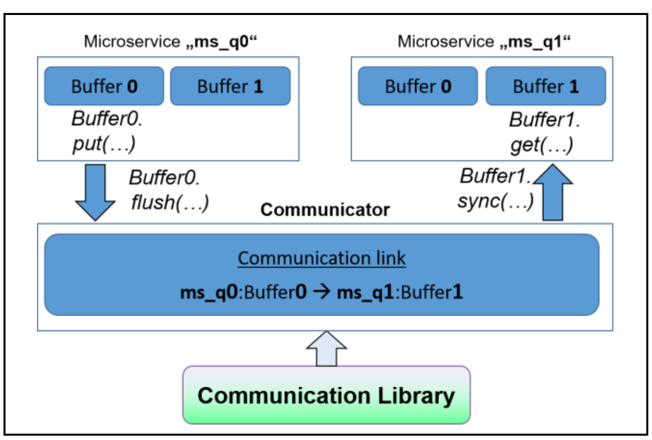


Figure 12. Realization of indirect communication between microservices.

The major dilemma, which nearly all microservice applications are facing during their design, is the definition of the granularity level of the initial problem that shall be implemented as a microservice. For example, in our case of the CFD applications, the lowest granularity level is served by a discretization element, and the highest – by a VN, see Fig. 7 above. In general, the lower is the chosen granularity level of the individual MS, the more customization options can be applied to the design of the MS-based application. However, the disadvantages of the very low chosen granularity level are getting very clear when the number of microservices in the MS-application is getting fairly big and the application has to spent most of the time in struggling with the management and coordination issues arising across the huge amount of very small MS. The "golden middle" strategy, which is used by some MS-based applications, is, however, not always applicable. On the contrary to the existing approaches, the MS-Sim design allows the developer (in our case – the simulation expert) to incorporate MS of different levels granularity in a single application, depending on the imposed customization and usability requirements. This hierarchical view is becoming possible to achieve by using the MS-Sim *hierarchy model* – a special methodology for creation of compound, hierarchically structured services. The MS-Sim hierarchical service concept allows not only to use the microservices of the different granularity in a common application, but also to create and reuse new, higher-level services on the base of the already existing ones. The developer will obtain the possibility to reuse services of any granularity level from a common "service market place". Fig. 13 shows an example of a microservice-based implementation of the U-formed coalmining section (depicted in Fig. 3) that combines the services of: i) discrete units (q, p) and ii) airways (Q, P) granularity level.

A prototype implementing the basic functionality of the *MS-Sim* framework was implemented with *OpenMPI* (see Gabriel et al. [17]) – an implementation of the Message-Passing Interface (MPI) standard, serving in our implementation as both a run-time environment and a communication library. *OpenMPI* is currently one of the most widely used solutions for designing parallel applications, mainly for HPC and cluster infrastructures. However, it is also useful for the implementation of MS-based applications: Same as the typical MPI applications, the SOA-based applications consist of a large number of components (microservices, in our terminology, or ranks, in the MPI terminology), which are running on distributed resource in isolated environments (as a software process). The components (regardless of whether they are microservices or MPI ranks) need to occasionally communicate between a couple of those (sender and receiver) or involving a group of components (i.e., collectively). In *MS-Sim*, every microservice is wrapped into an MPI process and is identified, in addition to its own Identifikator, by an MPI rank. The *communication library* (see Fig. 12) substitutes the data transfer calls with a corresponding MPI library calls, such as asynchronous point-to-point or collective communication.

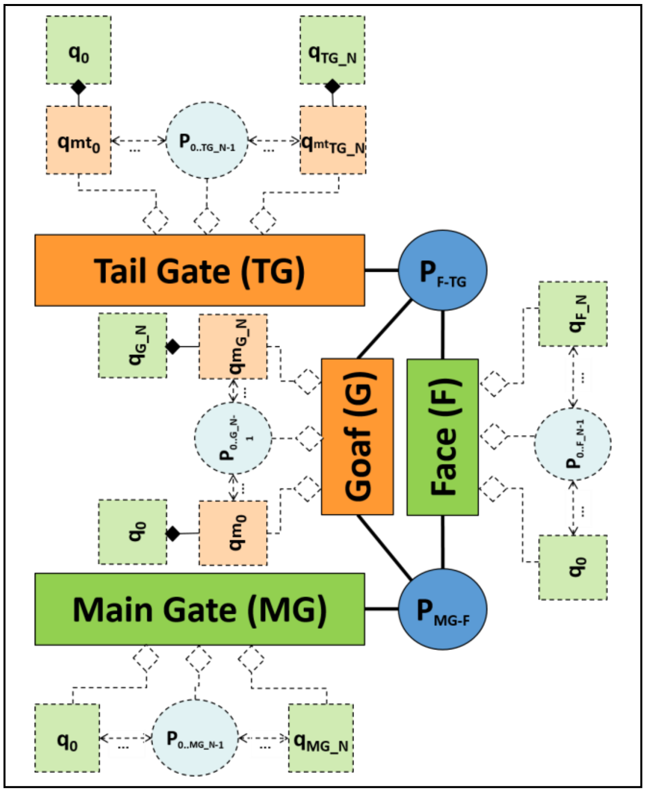


Figure 13. Implementation of ventilation section model with microservices of different granularity levels.

V. VALIDATION AND BENCHMARKING RESULTS

The ventilation section model, depicted in Fig. 12, which includes the models of the air-flow through the *main gate* and *face*, air leakage into the filtration cavity of the *goaf* and methane emission, as well as transport of the air-methane mix through the *tail gate* was evaluated for a serious of typical changes that might be applied to the operational mode of the VN regulation elements – the main *ventilation fan* and local regulator (a *sluice valve* installed in the *tail gate*). The discretization of the physical domain was performed with the help of method of lines (NMOL), as served by the equation (6), with the approximation element length of $\Delta\psi$=50m. The numerical solution (time discretization) was performed with Runge-Kutta (RK4) method with the time stamp of 0.1 s. The service-based application workflow was steered by a dedicated supervising service – a manager, which performs the following functions:

- Initialization of microservices with the parameters that correspond to their respective physical object.
- Initiation of iterative numerical solution process.
- Control of the solution readiness by means of polling the status of every individual “worker service”.
- Instruct the services to store results in a local storage (the local disk).

Goal of the evaluation was: i) to validate the functionality of the models as compared with the real measurements from the test objects and ii) to benchmark the performance of simulation services in order to evaluate the applicability for real-time and HPC/Cloud usage scenarios.

The **functional validation** was performed on the basis of a test object (a section of the coalmine South-Donbass-3) for three subsequent regulation scenarios (*A*, *B*, *C*):

- substantial increasing of the ventilation fan pressure *H* from the value *0* to the full capacity $160{,}69\ N/m^2$ (scenario A),
- decreasing of the air throughput by means of raising the regulation element’s value *r* from *0* to $25\ Ns^2/m^8$ (scenario B), and
- returning of the regulator in the initial position (scenario C).

The values of the airflows that resulted from simulation (see Table 1) were validate across the practical measurements data that were obtained by Svjatnyj [8]. The validation has not found any deviations between the modelling experiments and real measurements.

The **non-functional performance benchmarking** was performed for two following scenarios:

- Performance on a range of non-parallel systems: an embedded device (a low-power ARM-based embedded system Odroid-XU4 [18]), a generic desktop PC (Intel Core i5 CPU), and a server system equipped with a high-performance Intel Haswell CPU. The evaluated figure of merit was the ability to meet real-time requirements, i.e., $T_{sim} < T_{phys}$, where T_{sim} – the simulation time (duration of simulation experiment execution), T_{phys} – duration of dynamic process in real-time of the physical process. Only *main gate* model was used in this benchmark.
- Performance on a parallel (HPC/Cloud) infrastructure. The evaluated figure of merit was the parallel efficiency, measured as $T_{Sim}/(N_{Cores}*T_{Ncores=1})$, where N_{Cores} – the number of parallel processing units (e.g., CPU cores), $T_{Ncores=1}$ – the simulation time on a single processing unit. The evaluated system was served by a cluster with Intel Haswell CPU and Infiniband interconnect.

The performance results on non-parallel systems are shown in Fig. 14. In order to test the scalability on different hardware architectures, the evaluation was repeated for a doubled set of microservices (also involving more communication between them), which was achieved through refining the discretization mesh from $\Delta\psi$=50m. (corresponds to 8 microservices representing the modelled main gate) to $\Delta\psi$=25m. (i.e.,16 microservices, accordingly).

Benchmarking for the test case have revealed, that all systems were able to satisfy the real-time requirements (by the measured physical duration of the dynamic process of approximately 3 hours). As expected, the worst scalability results (decrease of performance by increased simulation’s computation load) were shown by the low-performance embedded system, however, without any implication on the usability in real-time scenarios.

TABLE I. VALIDATION RESULTS FOR TEST OBJECT

Parameter	Experiment (A)		Experiment (B)		Experiment (C)	
	Start	Finish	Start	Finish	Start	Finish
Regulation Actions						
Local regulator resistance **r´** *[Ns²/m⁸]*	0	0	0	**25**	25	**5**
Ventilator depression **ΔP** *[N/m²]*	0	**160.69**	160.69	160.69	160.69	160.69
Measurement Values						
Airflow **Q** *[m³/s]*						
- Main Gate	0	7.2	7.2	2.38	2.38	4.45
- Face	0	5.6	5.6	1.86	1.86	3.45
- Goaf	0	1.6	1.6	0.52	0.52	1.00
- Tail Gate	0	7.2	7.2	2.38	2.38	4.45
Simulation Results Values						
Airflow **Q** *[m³/s]*						
- Main Gate	0	7.199	7.199	2.39	2.39	4.45
- Face	0	5.599	5.599	1.86	1.86	3.46
- Goaf	0	1.599	1.599	0.53	0.53	0.99
- Tail Gate	0	7.199	7.199	2.39	2.39	4.45

The important outcome of the nonfunctional properties evaluation is the fact that despite of being extremely compute time consuming and communication intensive, the CFD models of ventilation networks can be executed even on relatively small and with weak performance low-power systems, which are interconnected by means of commodity networks, such as Ethernet. Of course, imposing the necessary level of approximation (as in our case that deals with 1D-models) is the major prerequisite for the ability of such small systems to carry out the simulation successfully; however, the precision of the obtained simulation results allows the simulations to be used for the real cases in production without any considerable restrictions.

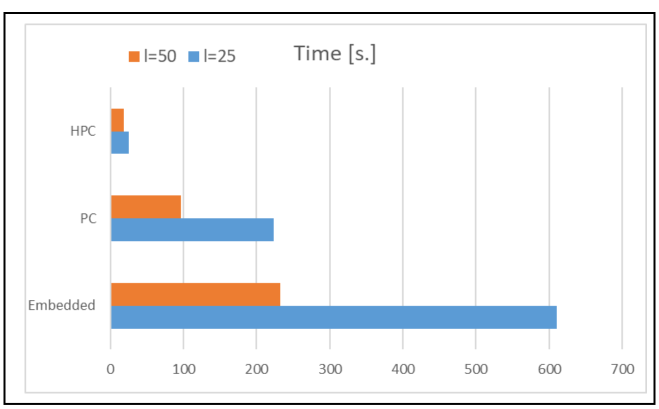


Figure 14. Performance results on diverse hardware architetures.

As for a highly optimized parallel high-performance system, the scalability of the applications is one of the major criteria for the usability of parallel applications. This characteristic was tested on a parallel cluster with Infiniband interconnect and Intel Xeon processors. The parallel efficiency that was obtained on the cluster is shown in Fig. 15. The implemented benchmark shows very positive results (efficiency over 100%) for up to 72 cores. Afterwards, the efficiency starts lowering, due to undersubscription of the CPU cores, as the number of microservices remains unchanged. Overall, according to the evaluation results, the pilot implementation of the *MS-Sim* platform, supported by *OpenMPI*, was able to achieve good performance and can be applied in both real-time and HPC simulation cases.

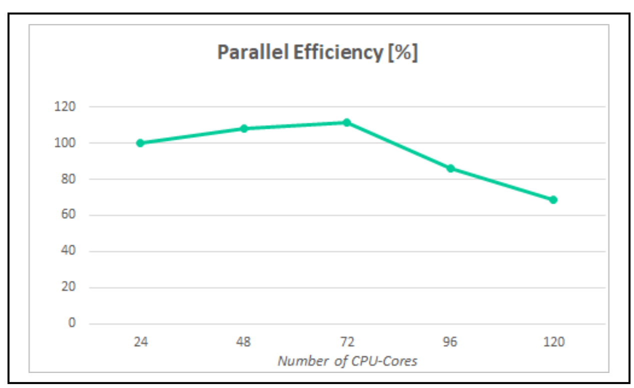


Figure 15. Parallel efficiency on cluster.

VI. Conclusion

The simulation technology is facing the challenges of application for new real-time scenarios that require a high flexibility of modelling tools in terms of the broader usage of the available infrastructure (data acquisition, storage, and processing devices). The rapid development of sensor networks has made possible a number of new innovative scenarios, for which the monolithic design of the existing simulation tools and workflow solutions on their top might be of a big obstacle. Service-oriented platforms offer a promising vision of the future development of simulation tools by offering benefits of on-demand distribution and parallelization, which might be well supported the underlying management platforms. Microservices are the technology that can if not fully replace the workflow-based scenarios but have the potential to support and bring them to the principally new level of usability. The effort that was done on implementation of the ventilation scenario has revealed a high potential of microservices architectures in geoscience and other domains of science and technology.

The dynamic approach, in which the services act as independent interactive components that are continuously running on the dedicated hardware and can be steered by a remote controller according to the specific application logic is particularly interesting for real-time control scenarios. In such scenarios, the services can incorporate the sensor data, make prediction for the future situation development, and instruct the control system about the probability of any potential risks appearance. On the other hand, the models can be optimized by adapting their parameters to best fit the actual mode of the controlled complex dynamic system.

The further research related to the application of the microservice simulation concept to the ventilation networks (and other dynamic systems with topological structure) will concentrate on the following activities:

- elaboration of functional composition strategies to development of complex, assembled service for hierarchically-organized systems,
- evaluation of the energy consumption of microservices on the embedded systems and elaboration of strategies to its minimization,
- extension of the communication library to support other commodity data transfer libraries, such as any REST-based communication frameworks, such as *RabbitMQ*, etc.,
- development of interfaces to communicate with the industrial sensors (field bus),
- simplification of API towards automation of complex microservices creation,
- modularization of the numerical solvers' library, in order to allow a seamless selection of the required numerical method to be applied on the models.

The other important direction of work will be the adoption of the microservice programming concept and the implemented MS-Sim framework by the further application scenarios in the HPC community. The advantages that are offered by the SOA style of programming MIMD parallel applications offers advantages for many other application domains in the engineering and other scientific computing fields. The developed microservice-based programming technology is expected to be useful for the design of environmental control systems, biomedicine applications, traffic prediction and control systems, as well as many others.

Acknowledgment

The work presented in this paper has become possible thanks to the support of the EU project *ChEESE*, which has received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement Nº 823844.

References

[1] A. Cheptsov and O. Beljaev, “A Microservices Approach for Parallel Applications Design: A Case Study for CFD Simulation in Geoscience Domain”, In proc. GEOProcessing 2020: The Twelfth International Conference on Advanced Geographic Information Systems, Applications, and Services, pp. 25-30, 2020.

[2] B. Bordel Sánchez, R. Alcarria, and T. Robles, “Cyber-physical systems: Extending pervasive sensing from control theory to the Internet of Things”, Pervasive and Mobile Computing, 2017.

[3] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0”, IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17-27, 2017.

[4] C. Stewart, S. Aminossadati, and M. Kizil, “Use of live sensor data in transient simulations of mine ventilation models”, Mining Report 153 (4/2017), pp. 356-363, 2017.

[5] Internationales Energy Programm (IEA), “World Energy Balances 2019”, [Online]. Available from: https://www.iea.org/statistics/coal/. Last access on 5.12.2020.

[6] Wikipedia. Statistics of major catastrophes in coal mines, [Online]. Available from: https://de.wikipedia.org/wiki/ Liste_von_Ungl%C3%BCcken_im_Bergbau/ 2020.03.24. Last access on 5.12.2020.

[7] A. Chorin and J.-E. Marsden, “A Mathematical Introduction to Fluid Mechanics”, Springer Verlag, 2000.

[8] V. Svjatnyj, “Simulation of aerodynamic processes and development of control system for underground mine ventilation”, PhD thesis 1985, (in Russian).

[9] L. Feldmann, “Dynamics analysis and automated control systems synthesis for coalmine ventilation”, PhD thesis, 1974, (in Russian).

[10] H.J. Bungartz, S. Zimmer, M. Buchhoz, and D. Pflüger, “Model creation and simulation. A practise-oriented guide”, Springer Verlag, 2013, (in German).

[11] R. Rannacher, “Finite element methods for Navier-Stokes equation”, Lecture notes on theoretical and numerical fluid mechanics in Vancouver, 2019.

[12] L. Feldmann, V. Svyatnyy, M. Resch, and M. Zeitz, “Research domain: Parallel simulation technology”, In proc. ASIM: The 3. International Conference on Simulation and Computer Graphic, pp. 139-168, 2009.

[13] E. Lee, “The past, present and future of cyber-physical systems: A focus on models”, Sensors 15(3), pp. 4837-4869, 2015.

[14] J. A. Kay, R. Entzminger, and D. Mazur, "Industrial Ethernet - overview and best practices", Conference Record of 2014 Annual Pulp and Paper Industry Technical Conference, Atlanta, GA, pp. 18-27, 2014.

[15] N. Ignatovych, “Perspectives of implementation of automated contol system UTAS in coal-mines”, Problems of work safety in Ukraine, 17(2009), pp. 3-16, 2009, (in Ukrainian).

[16] MS-Sim reference framework implementationat Github, [Online]. Available from: https://github.com/alexey-cheptsov/MS-Sim. Last access on 5.12.2020.

[17] E. Gabriel et al., “Open MPI: Goals, concept, and design of a next generation MPI implementation”, In proc. 11th European PVM/MPI Users' Group Meeting, pp. 97-104, 2004.

[18] Pollin, Odroid-XE4 short specification, [Online]. Available from: https://www.pollin.de/p/odroid-xu4-einplatinen-computer-samsung-exynos-5422-2-gb-2x-usb-3-0-810409. Last access on 5.12.2020.

On Classifying Urban Issues from TV News Using Natural Language Processing and Geoprocessing

Rich Elton Carvalho Ramalho
Information Systems Laboratory
Federal University of Campina Grande
Campina Grande - PB, Brazil
rich.ramalho@ccc.ufcg.edu.br

Anderson Almeida Firmino
Information Systems Laboratory
Federal University of Campina Grande
Campina Grande - PB, Brazil
andersonalmeida@copin.ufcg.edu.br

Cláudio de Souza Baptista
Information Systems Laboratory
Federal University of Campina Grande
Campina Grande - PB, Brazil
baptista@computacao.ufcg.edu.br

Ana Gabrielle Ramos Falcão
Information Systems Laboratory
Federal University of Campina Grande
Campina Grande - PB, Brazil
anagabriellee@gmail.com

Maxwell Guimarães de Oliveira
Information Systems Laboratory
Federal University of Campina Grande
Campina Grande - PB, Brazil
maxwell@computacao.ufcg.edu.br

Fabio Gomes de Andrade
Federal Institute of Paraíba
Cajazeiras - PB, Brazil
fabio@ifpb.edu.br

***Abstract*—Citizens as sensors enable society to discuss urban issues. Although some geosocial networks have been developed in recent years to enable citizens to report many types of urban problems, the engagement of the users of these networks usually decreases in time. Hence, many relevant issues are not posted reducing the effectiveness of these networks. Aiming to overcome this limitation, this article proposes an approach in which urban issues are automatically detected from a television news program. The proposed solution uses geoparsing and Natural Language Processing techniques to geocode and classify the identified complaints. The results are published in the Crowd4City geosocial network that deals specifically with urban issues. Finally, our method was evaluated using data from a real news TV program in Brazil. Our results indicated 59.8% of success in extracting text and location from the video news.**

Index Terms*—*Geosocial network; NLP; Urban Issues; Crowdsourcing.

I. Introduction

The high concentration of population in urban areas has imposed on local authorities several challenges to address issues concerning mobility, security, infrastructure, education, health, etc. These are what we call urban issues. One important challenge for these authorities consists of identifying the problems that have been faced by citizens.

Aiming to solve this limitation, in the context of Smart Cities, some authors have developed geosocial networks that deal specifically with urban issues. These networks enable the use of context-aware services to locate users and their complaints.

In a previous work [1] we provided a discussion towards the proposal for an approach for automated detection of urban issues from TV news programs. In this work, we improve our previous proposal defining domain ontology named UIDO (Urban Issues Domain Ontology), which aims to model the semantics of urban issues. We also extend the case study with a performance analysis in order to compare our classifier with XGBoost and Bi-LSTM classifiers. Finally, this work also provides a state-of-the-art comprehensive review in the field.

In the context of Smart Cities, geosocial networks enable the use of context-aware services to locate users and their complaints about urban issues. Several tools, such as Crowd4City [2], Wegov [3], and FixMyStreet [4] provide urban complaint environments. However, people's motivation to use such geosocial networks decreases throughout time. Hence, to ensure a high engagement of society, different approaches to gathering information are required.

Several local TV stations in Brazil portray urban issues addressed by the community. An example is the 'Calendar' report in a daily open TV channel news program in the State of Paraíba, Brazil. That news broadcast exhibits several urban issues from the main cities of that particular state. Then, we decided to use the information presented in this broadcast to automatically input new urban issues into geosocial networks, improving citizenship and increasing awareness. To accomplish this task, we initially convert the audio descriptions in the news channel into text. After that, we use geoparsing tools from Geographic Information Systems (GIS) and Natural Language Processing (NLP) techniques to extract the correct location of the respective urban issues.

In this article, we propose a framework to extract audio files from TV news, convert them into text documents, then extract location using a gazetteer and urban issues from text using NLP techniques in order to feed the Crowd4City geosocial network. It is important to mention that the news is up-to-date and extracted from a real context. Our main contribution consists in the integration of GIS and NLP.

The remainder of this article is structured as follows. Section II discusses related work. Section III presents an overview of the Crowd4City geosocial network. Section IV focuses on our proposed method for extracting and structuring urban issues reported in TV news. Section V presents a case study and

The authors would like to thank the Brazilian Research Council - CNPq for funding this research.

discusses the results. Finally, Section VI concludes the paper and points out further research to be undertaken.

II. Related Work

NLP techniques have been broadly used in several application domains including machine translation, speech recognition, chatbots/question answering, text summarization, text classification, text generation, sentiment analysis, recommendation systems and information retrieval. Britz et al. [5] discuss machine translation using a seq2seq model. Reddy et al. [6] present a question answering approach. Schwenk et al. [7] focus on text classification. Radford et al. [8] propose a language model using unsupervised learners. NLP is difficult to accomplish since text differs from language to language.

Upon developing our proposed approach, we first performed an extensive study on the already existing models within scenarios similar to ours. Given that our method is based on NLP and geoparsing, we discovered some useful corpora. Oliveira et al. [9], for instance, contributed to the creation of a gold-standard english corpus of urban issues identified from geo-located tweets. Such information can be very useful for improving geoparsers and for developing classifiers for the detection of urban issues. Focusing on our TV news domain, Camelin et al. [10] composed a corpus of different TV Broadcast News from French channels and online press articles. These articles were manually annotated in order to obtain topic segmentation and linking annotations between topic segments and press articles. The FrNewsLink is freely available online. Although both corpora are based on different languages than the one used in our study, they proved to be useful in such domains.

Aiming at obtaining information from the TV news videos, Kannao and Guha [11] focused their study on extracting text from the overlay banner presented in broadcasts. Such text usually contains brief descriptions of news events. They performed experiments using Tesseract Optical Character Recognition (OCR) for overlay text recognition trained using Web news articles. The authors validated their approach using data from there Indian English television shows and obtained significant results. However, their domain is limited. In a more recent study [12], the authors proposed a system architecture that enables the semantic segmentation of TV news broadcast videos. In this architecture, the collected videos were segmented into shots from which advertisements were detected and removed. Then, the remaining non-commercial content was classified as news bulletins, debates, or interviews. Finally, contents categorized as news bulletins were processed aiming at obtaining news stories. To validate the proposed architecture they used videos from three Indian English news channels. Nonethetess, they do not address urban issues.

Chifu and Fournier [13] developed the SegChainW2V framework. In their work, video segmentation was accomplished from lexical chains obtained by transforming the results of video transcriptions into vectors. Then, they used cosine-based similarity measures to detect topic variations along with a video and applied the word2vec word embeddings model for computing similarities between the videos. They used data from a French TV news broadcast and an English MOOC video, and the preliminary results showed the viability of the proposed solution. Nevertheless, unlike the solution described in this article, this work did not deal with urban complaints.

Iwata et al. [14] enabled users to retrieve news videos from keywords. In their solution, the author used OCR for extracting captions from video images in Arabic. Then, they applied of text processing techniques to perform character recognition and automatic language translation. Their approach aimed at evaluating the use of frame images extracted from the AlJazeera broadcasting programs. Despite the relevance of this work, it does not evaluate the semantics of the videos.

Similarly, Pala et al. [15] developed a system for the transcription, keyword spotting and alerting, archival and retrieval for broadcasted Telugu TV news. Their main goal was to aid viewers in easily detecting where and when topics of their interest were being presented on TV news in real-time and they were also hoping to assist anyone (including editorial teams at TV studios) in discovering videos of TV news reports about specific topics, defined by the user with keywords. Their system was the first that enabled the simultaneous execution of the broadcasted audio (speech), video and transcription of the audio in real-time with the Indian Language, with keyword spotting and user alerts. Although it can detect topics of interest based on keyword, the system cannot identify the theme to extract the theme or domain being discussed in the video.

Bansal and Chakraborty [16] proposed an approach for content-based video retrieval by combining several state-of-the-art learning and video/sentence representation techniques given a natural language query. They aimed at overcoming the robustness and efficiency problems found in the existing solutions using deep-learning based approaches, combining multiple learning models. Their results showed that their solution was able to capture the videos' and sentences' semantics when compared to other existing approaches. However the authors lack retrieving any geographic information.

Dong et al. [17] focused on developing a method for subject words extraction of urban complaint data posted on the Internet. Their approach consisted in the segmentation of the complaint information, extraction and filtering of candidate subject words, and was validated using 8289 complaints posted on a Beijing website. The proposed method showed that better results can be obtained than the Term Frequency–Inverse Document Frequency (TF-IDF) and TextRank methods in the context of written informal content made by Internet users. Nonetheless, such an approach would need to be validated in other scenarios.

Mocanu et al. [18] proposed a method for news extraction by using temporal segmentation of the multimedia information, allowing it to be indexed and thus be more easily found the users interested in specific topics. Their approach was based on anchorperson identification, where the TV news program presenter was featured on the video. They performed some tests with a limited database of French TV programs. Nevertheless, the topic detection method implemented in this work is not very robust, since it is based only on the video subtitles.

Zlitni et al. [19] addressed the problem of automatic topic segmentation in order to analyze the structure and automatically index digital TV streams using operational and contextual characteristics of TV channel production rules as prior knowledge. They used a two-level segmentation approach, where initially the program was identified in a TV stream and then the segmentation was accomplished, thus dividing the news programs into different topics. They obtained reasonable results in their experiments, but their approach is completely dependent on the production rules of TV channels. Also aiming at achieving news story segmentation, Liu and Wang [20] focused their efforts on using a convolutional neural network in order to partition the programs into semantically meaningful parts. They based their input on the closed caption content of the news and trained and tested their model using the TDT2 dataset, from Topic Detection and Tracking (TDT). Although they obtained significant results, their approach is limited to the linguistic information extracted from the closed caption and thus it is not applicable to programs without such resource.

Even though several studies could be found, none of them comprises the same aspects and goals we aim at achieving with our study, which is to perform NLP and GIS extraction and structuring of stories depicted in TV news reports, focusing especially on urban complaints.

III. Ontologies versus Topic Modeling for urban issues automated extraction

The classical definition of ontology is provided by Gruber [21]: "An ontology is a formal explicit specification of a shared conceptualization". In other words, ontologies provide a shared vocabulary that can be used to model a domain. Such a shared vocabulary can be represented by objects and/or concepts that may contain properties and relationships [22].

In computer science, ontologies have been used since the last decades for accomplishing several tasks such as improving communication between agents (human or software); enabling computational inference; and reusing data models or knowledge schemas [23]. In this context, ontologies can be classified according to the language expressivity and scope. Language expressivity focuses on knowledge representation. Such a classification includes information ontologies, linguistic ontologies, software ontologies and formal ontologies. On the other hand, the scope addresses the level of specificity of the knowledge represented by the ontology and includes domain ontologies and general ontologies.

Concerning language expressivity, information ontologies are a clarified organization of ideas useful solely by humans. Linguistic ontologies can be dictionaries, folksonomies [24], lexical databases, etc. One known information ontology example is the Resource Description Framework (RDF). Software ontologies focus on data storage and data manipulation for data consistency in software development activities. The Unified Modeling Language (UML) is one example of a software ontology. Formal ontologies require some clear semantics and involve formal logic and formal semantics, with strict rules about how to define concepts and relationships. The Web Ontology Language (OWL) a widely used language that enables people to build formal ontologies. Finally, concerning the scope of this article, domain ontologies stand out. Different from general ontologies, domain ontologies focus on specific domains, such as urban issues, with specific viewpoints and characteristics, to better represent the domain semantics.

A way of extracting valuable information from texts is using Ontology-based Information Extraction (IE), as performed by Yang et al. [25]. Ontology-based IE processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information [26]. Two main categories split ontology-based IE: document-driven and ontology-driven. Document-driven [27] is known as semantic annotation, where the discovered knowledge is structured in domain ontologies, while ontology-driven [28] extracts information from unstructured documents based on a domain ontology constructed under the help of domain experts or even by combining domain ontologies with core reference ontologies and folksonomies.

While domain ontologies represent structured and specialized semantic knowledge, topic modeling is the discovery of hidden semantic structures in a text [29]. In this context, unsupervised topic modeling is an unsupervised machine learning task that can scan a set of documents to detect word and phrase patterns a set of documents, detecting word and phrase patterns within them, being suitable for dealing with a large-scale dataset [30]. Topic modeling also has the capability of automatically clustering word groups and similar expressions that best characterize a set of documents. Thus, topic modeling can be seen as a baseline to build domain ontologies in order to enable performing ontology-based IE in specific domains such as urban issues.

In the context of the urban issues domain, this article applies topic modeling combined with location extraction in order to identify not solely the main keywords for this domain, but also the hidden semantics of urban issues. Such hidden semantics may include the way of people complain about urban issues such as potholes, broken lights, litter and others. This is a key step towards building a domain ontology in the urban issues domain that can be applied to perform ontology-based IE in unstructured texts from social media. Moreover, once a domain ontology in urban issues is built, it can be useful to many other tasks. One of the main advantages in using ontologies is the interoperability by humans and the ability to perform inferences, which makes the domain modeling easier and readable by machines. In order to illustrate how ontologies can be used to model semantically urban issues, in the following we present a domain ontology for urban issues.

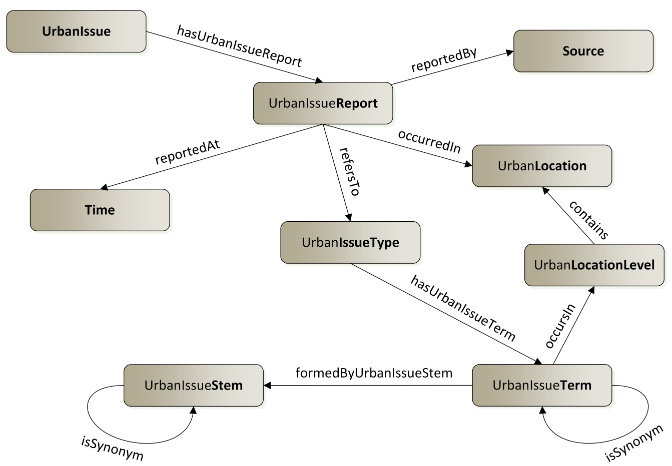


Fig. 1. Main concepts and relationships from the UIDO ontology

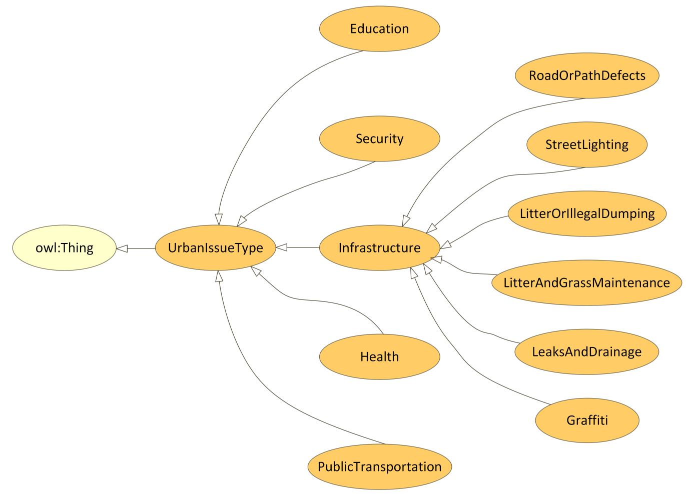


Fig. 2. The urban issue type hierarchy

A. *The UIDO ontology*

We designed the UIDO - Urban Issues Domain Ontology for semantic modeling of urban issues [9], [31]. The design phase for the UIDO ontology started by analyzing the CContology [32], a domain ontology developed to support online customer complaint management. Such ontology contains seven main classes: Complaint Problems, Complaint Resolutions, Complaint, Complainant, Complaint-Recipient, Address, and Contract. From these, only four classes would be relevant to create the UIDO ontology: Complaint, Complainant, Complaint Problems and Address. However, these four concepts could not be directly reused because they focus on specific customer complaints that are quite different from the complaints in the urban issues domain. Therefore, only the CContology design ideas were used for defining the concepts and relationships for the UIDO ontology.

The UIDO ontology was developed using the Web Ontology Language (OWL) [33] using the Protégé 5.0 system. The core of the UIDO ontology is the classes UrbanIssueReport. this class has two main goals: to act as a top concept during the reasoning process on discovering urban issues from a text; and to be a metamodel for the Linked Data entities from the social media messages related to urban issues. Figure 1 shows the main concepts and relationships of the UIDO ontology. These concepts and relationships make the basic skeleton, but there are others in the sublevels of the hierarchy.

The classes UrbanIssueType and UrbanLocationLevel (shown in Figure 1) are explained in more detail because they are the most important concepts beyond the core. The core class *UrbanIssueReport* represents the reports regarding urban issues. The "user" attribute is encapsulated into the *Source* class as well as the "description" attribute. The *UrbanIssue* class groups urban issue reports of the same type, location and time by the *hasUrbanIssueReport* relationship.

The *UrbanIssueType* class models the classification of issues related to the urban environment. There are initially five subclasses for urban issue types: education, security, infrastructure, health and transportation. Figure 2 shows the UrbanIssueType hierarchy. The six issue types learned from the FixMyStreet corpora fits in the infrastructure concept. Thus, they are included as *Infrastructure* subclasses. Other corpora and specialist knowledge would be necessary in order to expand the ontology within other classes and subclasses that may be further included.

Figure 3 shows an example of two instances ("broken bollard" and "pavement damage") from the classes *UrbanIssueTerm* and *UrbanIssueStem* in a relationship with the issue type "Road or Path Defects" from the *UrbanIssueType* class.

The *UrbanLocationLevel* class models the level of detail for geographical locations. Figure 4 shows the urban location level hierarchy. Such levels are used in constraints for urban issues locations according to the urban issue term. These constraints are modeled through the relationship *occursIn*. For example, a detected urban issue report about a pothole is relevant if it occurred at street or POI (Point of Interest) levels. However, at higher levels such as city or county, such a report tends to be irrelevant due to the geographic vagueness. There are initially three subclasses: Street/Road, District and POI. Nevertheless, other levels may be further included according to new studies. The semantics of such instances are explained in details in the following section.

The *UrbanLocation* class models the geographical information of urban issue reports. Such class reuses concepts from the LinkedGeoData ontology (LGDO) [34]. The LGDO ontology was chosen for the spatial locations related to urban issues for the following reasons: 1) It is derived from concepts defined by the OpenStreetMap [35]; 2) OpenStreetMap is currently a huge up-to-date and open spatial database that contains spatial features, mostly inside urban areas, such as street networks, which are commonly found in urban issue reports; 3) A preliminary work (presented in the following section) that focused on the geographical facet of urban issues enables to store spatial features from OpenStreetMap and provides the

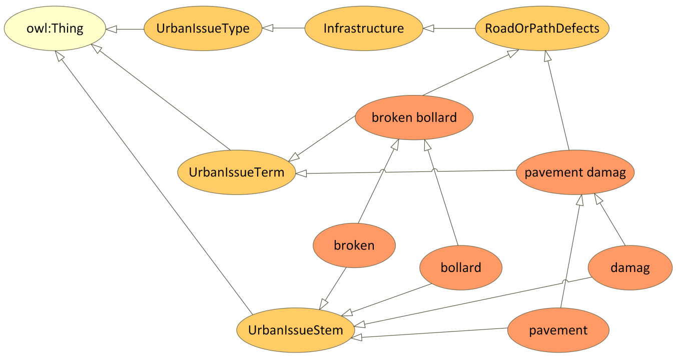


Fig. 3. An example of urban issue stems and terms related to an urban issue type

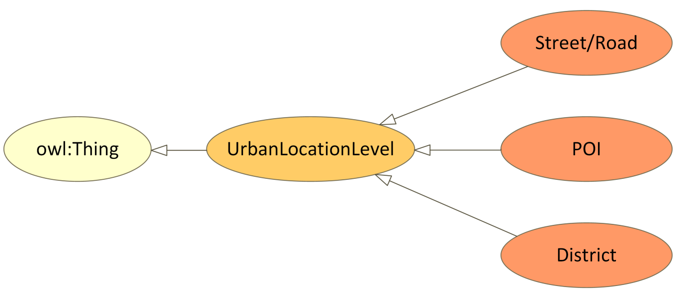


Fig. 4. The urban location level hierarchy

location levels for the UIDO ontology.

The *Time* class models the temporal information of an urban issue. The temporal information can be assigned to the time which the social media message was posted or it can be extracted from the social media message using temporal parsers. Such class reuses concepts from the OWL-Time ontology [36]. The OWL-Time ontology models temporal concepts describing the temporal properties of resources in the world or described in Web pages such as urban issues reported in social media. Thus, the class *Time* in the UIDO ontology is equivalent to the *TemporalEntity* in the OWL-Time.

Finally, the classes *UrbanIssueTerm* and *UrbanIssueStem* model the key terms learned during the knowledge acquisition stage. These concepts are in charge of connecting the top concepts modeled with the terms learned from the analyzed corpus. A set of relevant stems are instances of the urban issue stem concept instead of words (elements from W_{ui}) as a stem is the root of a word and thus it better represents a group of words with similar semantics. The combined stems produce a set of relevant terms T_{ui} through the relationship *formedByUrbanIssueStem*. These terms are instances of the urban issue term concept. Then, subsets of these relevant terms compose the vocabulary of urban issue types through the relationship *hasUrbanIssueTerm*. Both term and stems can be synonyms for each other. For those cases, the relationship *isSynonym* applies.

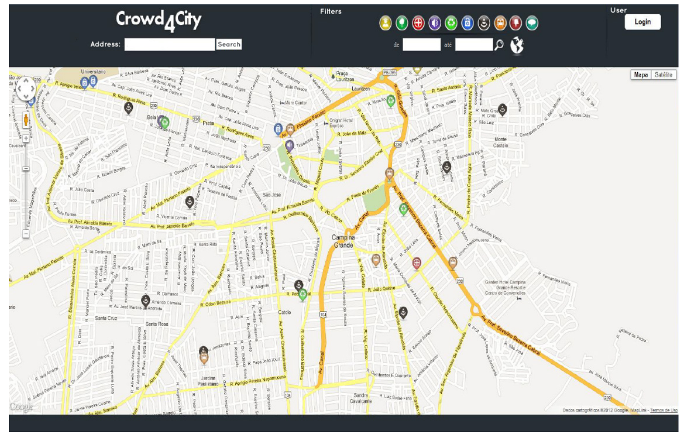


Fig. 5. Crowd4City's main user interface.

IV. THE CROWD4CITY GEOSOCIAL NETWORK

The Crowd4City system is a geosocial network aiming at providing e-participation to citizens, which enables them to take part more actively in their city's management, acting as sensors. The Crowd4City users can share and comment on many kinds of geolocated urban issues including traffic jam, criminality, potholes, broken pole lights and so on. Citizen's complaints about urban issues are shared publicly in the Crowd4City aiming to draw the attention of the authorities and the society as a whole. Hence, Crowd4City enables humans as sensors in a smart city environment. Figure 5 depicts the Crowd4City interface in which users can see the spatial distribution and pattern of different topics related to urban issues.

Regarding Crowd4City's use (Figure 5), the citizens can create complaint posts using their personal information or even anonymously, and they can input their dissatisfactions making use of the geographical tools. They can mark a single point on the map where the problem took place (for instance, if the user is reporting a pothole on a street); they can draw lines, perhaps to show routes where there are lighting issues; or they can even draw polygons on the map, thus being able to report regions that can be considered insecure.

Crowd4City presents some predefined categories for the problems reported including: Education, Sanitation, Transportation, Work Under Construction, Security and Others (Noise Pollution, Rubbish, Lighting, Potholes, etc.). However, if the user wishes to report something else, there is a category named "Other", which can be used for such uncategorized complaints.

Crowd4City's posts consist mainly of a location (geographic information), a title, a brief description and optionally multimedia attachments if the user has pictures or videos of the problem being reported. Additionally, the system provides a section for the other users's feedback with like/dislike buttons and a comment section, as seen in Figure 6.

Crowd4City enables operations such as pan and zoom. Also, the system provides several filters so that the users may perform more specialized searches for their information of interest. There are the basic filters, where the posts may be refined by the selected categories or creation dates; and the advanced filters, which may consider the complaints' contents and their geographic information. With the advanced filters, the users may perform searches using the buffer and contains operations, select some Points Of Interest (POI) categories (such as schools, hospitals, squares, airports and so on) and combine all the available filters.

A serious problem when dealing with information submitted directly from the internet users is the reliability of such data. Malicious users and spammers may use this tool to post irrelevant, erroneous or misleading information or even harmful URLs that can affect the users' devices. In order to bypass this problem, we elaborated a reputation model to be implemented in the Crowd4City geosocial network that evaluates every information submitted to the system and allows the users to better understand if what they are reading could be trustworthy or not. That way, we aim to provide a spam-free environment for Crowd4City, minimizing misinformation.

Our reputation model evaluates every action the users make, such as post creation and comments. That way, we assign a score to each user, which represents how trustworthy they are. When analyzing the content submitted in a new post or comment, the reputation model first checks if there are any of the flagged expressions (commonly used words associated with spam or malicious content, such as ”make money” or ”lose weight”, which were not expected to be used in the context of this geosocial network). According to the number of flagged expressions used, a spam score is attributed to it and that will have on user's reputation.

The second step adopted was the prohibition of uploading possibly harmful files, such as executables (.exe files). However, this is just a precautionary step and does not affect the reputation, since the file to be submitted could be.

We added in Crowd4City the possibility for the users to evaluate every post with ”I agree” or ”I disagree”. With

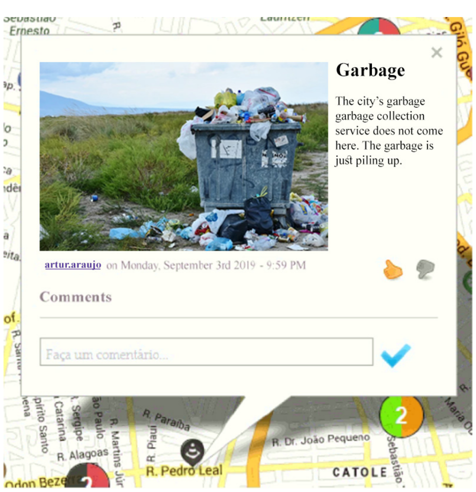


Fig. 6. A rubbish complaint.

the information retrieved from such feedback, we wish to determine if the content evaluated could be inappropriate, false or incorrect. Posts with a higher rate of ”I agree” feedback can indicate true and impartial content. The result of the feedback in a post created by the user affecteds his reputation score. However, the reputation of the user who sends a feedback is also taken into account, since a malicious user may try to ruin other users' reputation.

A more direct way for the users to express their discontentment with users they see that are not using Crowd4City properly is to report them as spammers on their profile pages.

Finally, one last feature is considered, which is the user participation in the system. Users that participate more actively creating post, commenting and evaluating other posts are more likely to have the system's best interests in mind, wishing for Crowd4City to achieve its full potential and be filled with truthful information that could be used to improve the city's conditions and its population's quality of life.

This way, users' reputation score is calculated considering: the reputation score associated with their posts and comments; the number of times they have been flagged as spammers; and how active they are. To each of these points, an expiration date is associated, in case of users change their behavior over time (once malicious users may change their mind and start using the geosocial network correctly).

As a result, this reputation score is shown graphically on the user's profile page with a red stamp (for users with low reputation scores), a white stamp (for neutral users), a bronze stamp, a silver stamp or a gold stamp (the last three for users with a good reputation, according to how good their scores are).

V. Automated Method for Extracting and Structuring Urban Issues Reported in TV News

The main problem addressed in this research deals with obtaining urban issues complaints from TV news, georeferencing them and automatically classifying them into one of the defined categories. The categories include sanitation, transportation, work under construction, among others. The urban issues context considered in this work is based on a corpus built in a previous work [9].

Our methodology comprises the following steps, according to Figure 7. First, we implemented a Web scraping method to extract the audio from video news. Second, we convert the audio into text using a speech recognition tool. Third, we use a gazetteer to perform geoparsing on the mentioned addresses and locations obtained from the Named Entity Recognition (NER) process, without preprocessing. Then, we implemented a preprocessing step comprising word capitalization, stopwords removal and lemmatization. Fourth, we use NER to obtain the named entities from the text. Then, we perform topic modeling to obtain the class of urban issues related to the text. Finally, the urban issues put into the Crowd4City geosocial network. We detail each step of our methodology next.

A. Web scraping - Video 2 Txt

Initially, we developed a Web scraping tool for obtaining the videos from a TV news website. The data comes from a Brazilian TV broadcast website in Portuguese. We used the Selenium library [37] and YouTubeDL [38] to download the audio files from the video URLs that were stored in a JavaScript Object Notation (JSON) file. Then, we used the SpeechRecognition library [39] with the Google Speech Recognition API to convert audio into text. In order to decode the speech into text, groups of vectors are matched to one or more phonemes, which is a fundamental unit of speech. The SpeechRecognition library relies on modern speech recognition systems based on neural networks and Voice Activity Detectors (VADs). In addition, Google Speech Recognition API is free and supports Brazilian Portuguese language with good results.

B. Preprocessing

In the preprocessing step, we converted the text into lower case, removed stopwords and performed lemmatization. We used the Spacy library [40] to perform entity recognition of locations. The Natural Language Toolkit (NLTK) Python library [41] was also used for the lemmatization process.

The strategy defined for the preprocessing was to extract words that are entities from locations using the Spacy NER, for which the library performs very well when aided by the SpeechRecognition tool. Spacy also offers support for the Portuguese language, which avoids the translation of all texts into English, as it may reduce performance.

Spacy recognizes the location entities of the text and their title, so we combine all the location entities found to then search for those addresses and choose the one with the highest reliability.

We also have guaranteed anonymization by removing the names of people that took part in the audio extracted from video URLs. Hence, privacy was preserved, although it is important to note that all the videos processed in this work are publicly accessible from the sources.

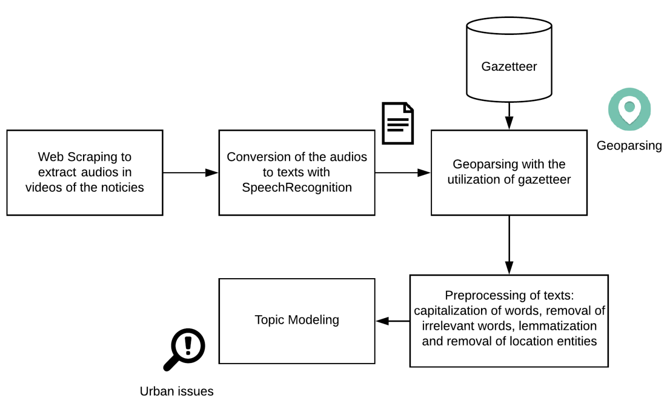


Fig. 7. Our proposed methodology.

C. Geoparsing

We used the Geocoder library [42], which offers an API that enables the use of geocoding services such as Google, ArcGIS, and Bing. The chosen API service was ArcGIS, which provides simple and efficient tools for vector operations, geocoding, map creation and so on. Brazil is ranked level 1 in the library, which means that an address lookup will usually result in accurate matches to the “PointAddress” and “StreetAddress” levels, which fulfills our requirements. After having the entities properly combined, we iterate through this structure by checking which one is the most accurate address (among the ones resulted by the Geocoder using ArcGIS). The accuracy is increased by filtering the addresses found, so the user may perform filtering by state, city or even geographic coordinate.

We used the Open Street Map (OSM) to obtain spatial data from some cities of the State of Paraiba in Brazil, and a gazetteer to improve the geoparsing accuracy. The gazetteer contains streets, neighborhoods, roads, schools, hospitals, supermarkets, pharmacies, etc. Notice that we do not deal with place name pronunciation, as the audio files do, because the names of the places were converted into text. We performed a cleaning of this data to keep only the information of interest to us: name, type and coordinates. Such cleaned data was stored in a PostgreSQL/PostGIS database system. Figure 8 presents the geoparsing step.

D. Topic Modelling

Concerning the topic modeling, we used Gensim [43], an open-source library for unsupervised topic modeling and NLP, which provides statistical machine learning tools. We used LDA from Gensim (LDAMulticore and LDAModel) to implement topic modeling, which considers each document as a collection of topics, and each topic as a collection of keywords.

In order to implement a topic classifier in Gensim, we needed to follow a few steps: creating both a word dictionary and a corpus (bag of words), then providing the desired number of topics and some algorithm tuning parameters. The word dictionary chooses an identifier for each words contained in documents. The corpus (bag of words) was a dictionary containing a set of the word identifiers and the number of occurrences of each word along the document. TF-IDF was also used, transforming the corpus co-occurrence matrix into a local TF-IDF co-occurrence matrix.

Concerning topic modeling, we removed all the words that were location entities, as they were not useful for the class classification process, aiming at increasing the accuracy of the model. Thus, our classifier focused on words of a given class, without worrying about locations.

To find out the best number of topics, some tests were performed and then we identified the best model comparing the evaluated models using the measure coherence score, which evaluates the quality of the obtained topics. After these tests, we concluded that the best number of data topics would be four, as shown in Figure 9.

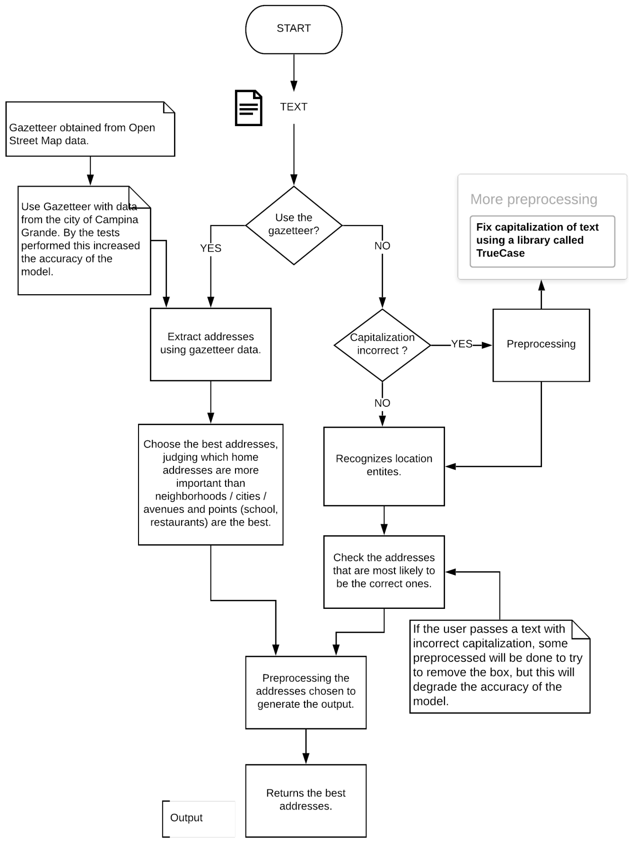


Fig. 8. The geoparsing process.

With four topics, the algorithm achieve a coherence score of 0.527613, the best result in the used dataset. Another improvement was the generation of the 15 most in the topics generated by the algorithm. After that, we manually selected the words that should not be considered and we added them to the list of stop words. Then, we repeated the process until the 10 words in each topic were strongly related to the topic.

In topic modeling, we can analyze which topics represent all documents and also the keywords of each topic. Figure 10 shows the thirty most frequent words in the first topic and also presents the words of the first topic sorted according to their relevance. The most important words were water, sewage, pavement, and home, thus indicating that the topic addresses sanitation problems.

VI. A CASE STUDY IN CAMPINA GRANDE NEIGHBOURHOOD

Usually, urban issues raises the attention of the local press, in order to establish a connection between population and city councils. In Campina Grande, a 400,000 inhabitants Brazilian city, citizens report their complaints to local TVs through messaging service platforms.

Thus, this research aims to fill the gaps mentioned above, helping to share complaints and providing a centralized means

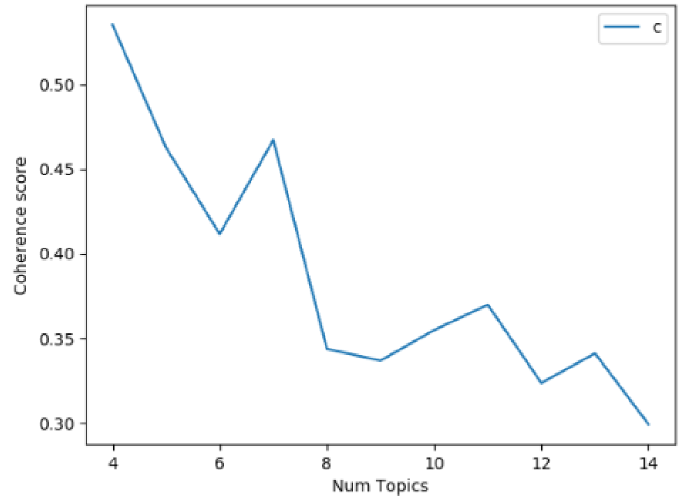


Fig. 9. Coherence score per number of topics.

with this information, it makes it easier for both the inhabitants to make complaints and for the local authorities to solve the reported problems.

A. Setup

In this research, we collected 1,007 videos of the news story "My Neighborhood on TV", covering the years 2016 to 2019, with an average duration of five minutes per video. We took all videos from the Paraiba TV news program website [44].

From all the videos obtained, in 602 of them (59.8 %) it was possible to obtain the text and the locations. Unfortunately,

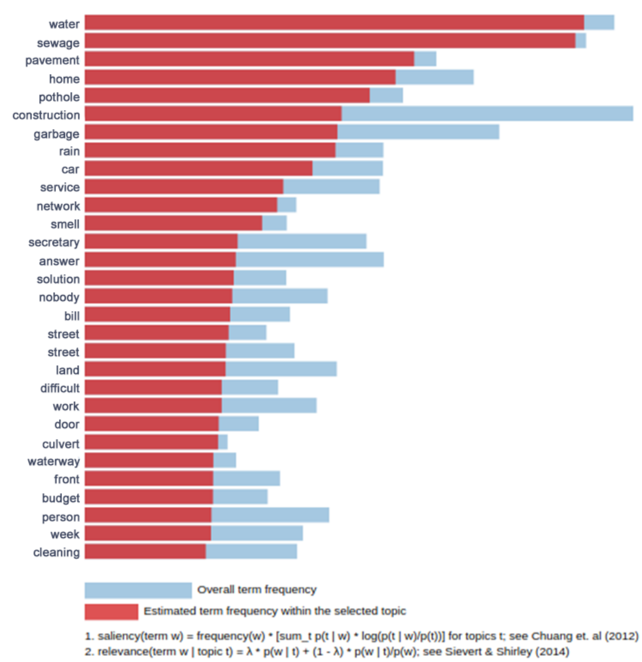


Fig. 10. Most frequent words in the first topic.

TABLE I
NUMBER OF TEXTS PER TOPIC

Class	**Number of instances**
Streets	308
Other (construction works)	236
Other (sanitation)	108
Education	91
Garbage	79
Other (traffic)	64
Complaints	26
Health	24
Security	17
Other (water shortage)	17
Public transportation	14
Forestation	5
Other	1

some videos did not specify the location. At the same time, some videos did not specify the urban issue, as the word "obra", which means "something not concluded that is being built or repaired" in Portuguese, is applied as a problem generalization.

We extracted the problem classes reported in the videos, enabling various applications to use them in an attempt to improve city management. Municipality authorities may be notified to provide solutions for urban issues.

When analyzing the data, we could see that there was a clear imbalance. Table I shows the arrangement of classes and the respective number of elements in each one.

As we can see in Table I, the majoritarian class had 308 elements, while there were many classes with less than 50 elements. However, as pointed by Branco et al. [45], when using unbalanced datasets, problems such as the use of biased assessment metrics to enrich the performance of the models in less frequent cases will occur. Also, it is necessary to force machine learning algorithms to focus on these less frequent cases.

Therefore, we carry out tests with all classes and decided to with performing use only the four classes with over 90 elements: Education, Other (sanitation), Other (construction works), and Streets.

B. Results

We performed several tests in the Gensim library, using different pre-processing functions and tests different values for their parameters. We set the number of steps equals to 10 because we saw that with this number we obtained better results without losing performance (see Figure 11). When we tried to use values below or above 10, the accuracy decreased.

We used Coherence Score to test the topics, which measures the relative distance between the words within a topic. The number of parameters used was 4, with a score of 0.52. Such a score is acceptable in this preliminary study due to our dataset. A problem using geoparsing is entity recognition. This is because the tools used for NLP cannot recognize entities that are misspelled (for example, if someone wrote the Campina Grande entity lowercase). However, this problem was mitigated with the use of the TrueCase library [46], which corrected the capitalization of words, so that the geoparsing could recognize the entities, obtaining good accuracy. It is important to notice that NLP is by definition not capable of getting the real meaning of any term or context, as text is something by nature completely different than language. In order to deal with context, we needed to combine NLP with other resources such as Part-Of-Speech tagging and supervised machine learning algorithms for instance.

The TrueCase library supports the English language only. As our data was in Portuguese, we needed to use a library to translate the words from Portuguese to English - GoogleTrans [47] - use the TrueCase and then do the reverse process, resulting the words in Portuguese with the correct capitalization.

As an additional process to improve the performance of geoparsing, we used a gazetteer, to improve in the geolocation process for texts of the city of Campina Grande - which is the object of this research.

Regarding topic classification, we analyzed the F-Score: 1) With all 13 classes and 993 complaints, and 2) Removing the classes with less than 90 instances, with 743 complaints remaining. In both cases, we carried out the same pre-processing (stemming, TF-IDF, etc.). Besides, we used two classification algorithms and compared their results. The classification algorithms used in this study were XGBoost [48] and a Bi-LSTM neural network (Bidirectional LSTM). [49].

Using XGBoost with all classes we obtained an a F-Score of 68.55%. After removing classes with few instances, we improved the results and obtained an F-Score of 83.34%. In Figure 12, we can see the F-Score with full data and only with the major classes. On the other hand, using the Bi-LSTM with all classes, we obtained an F-Score of 32%. With the removal of minority classes, the F-Score increased to 75%, as we can see in Figure 13.

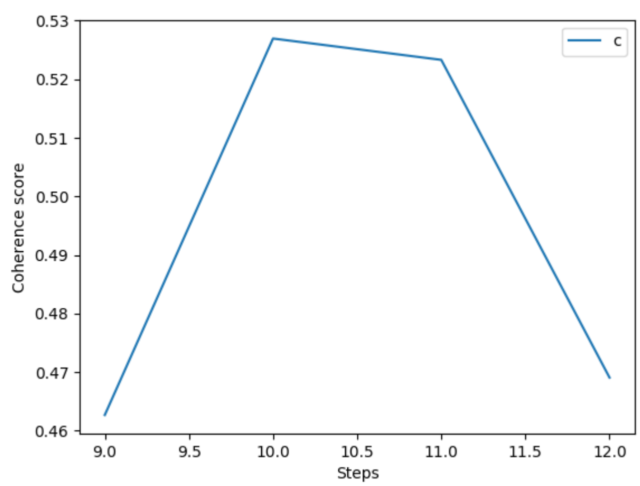


Fig. 11. Comparative chart for influence of the steps value in the coherence score.

Fig. 12. Comparison between the results with full data and with four classes using XGBoost.

Fig. 13. Comparison between the results with full data and with four classes using Bi-LSTM.

VII. Conclusion

Citizens as sensors enable the engagement of society through technology to complain about urban issues. Smart cities demand tools for such engagement promoting e-citizenship and e-participation. Nonetheless, although some of such tools have already been proposed, it turns out that people engagement decreases in time. Hence, the gathering of urban issues from any media is very important to keep people engaged. As such, this article proposes an approach to gather urban issues data from a TV news program using geoparsing and NLP techniques to locate and classify the urban issues in order to input it in the Crowd4City geosocial network.

The results showed that our approach is feasible and that we manage to classify urban issues into four topics: mobility, sanitation, buildings and others. As future work, we plan to perform an in-depth performance analysis of geoparsing, as well as topic modeling, by manually identifying the topics of the videos as ground truth and comparing them with the topic modeling results. Another plan consists of performing a comparative study between topic modeling and supervised machine learning.

References

[1] R. Rich Elton, F. Anderson, B. Cláudio, F. Ana Gabrielle, O. Maxwell, and A. Fabio, "Using Natural Language Processing for Extracting GeoSpatial Urban Issues Complaints from TV News." Valencia, Spain: IARIA, Mar. 2020, pp. 55–60.

[2] A. G. R. Falcão et al., "Towards a reputation model applied to geosocial networks: a case study on crowd4city," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC*, Pau, France, 2018, pp. 1756–1763.

[3] T. Wandhofer, C. van Eeckhaute, S. Taylor, and M. Fernandez, "We-Gov analysis tools to connect policy makers with citizens online," in *Proceedings of the tGovernment Workshop*, 2012, pp. 1–7.

[4] N. Walravens, "Validating a Business Model Framework for Smart City Services: The Case of FixMyStreet," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops*, 2013, pp. 1355–1360.

[5] D. Britz, A. Goldie, M.-T. Luong, and Q. V. Le, "Massive exploration of neural machine translation architectures," *ArXiv*, vol. abs/1703.03906, 2017.

[6] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," vol. Transactions of the Association for Computational Linguistics, Volume 7, March 2019, pp. 249–266. [Online]. Available: https://www.aclweb.org/anthology/Q19-1016 [accessed: 2020-03-02]

[7] H. Schwenk, L. Barrault, A. Conneau, and Y. LeCun, "Very deep convolutional networks for text classification." in *EACL (1)*, M. Lapata, P. Blunsom, and A. Koller, Eds. Association for Computational Linguistics, 2017, pp. 1107–1116. [Online]. Available: https://www.aclweb.org/anthology/E17-1104/ [accessed: 2020-03-02]

[8] A. Radford et al., "Language models are unsupervised multitask learners," 2018. [Online]. Available: https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf [accessed: 2020-03-02]

[9] M. G. de Oliveira, C. de Souza Baptista, C. E. C. Campelo, and M. Bertolotto, "A Gold-standard Social Media Corpus for Urban Issues," in *Proceedings of the Symposium on Applied Computing (SAC)*, ser. SAC '17. New York, NY, USA: ACM, 2017, pp. 1011–1016.

[10] N. Camelin et al., "FrNewsLink : a corpus linking TV Broadcast News Segments and Press Articles," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA), 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1329 [accessed: 2020-03-02]

[11] R. Kannao and P. Guha, "Overlay Text Extraction From TV News Broadcast," *CoRR*, vol. abs/1604.00470, 2016. [Online]. Available: http://arxiv.org/abs/1604.00470 [accessed: 2020-03-02]

[12] R. Kannao and P. Guha, "A system for semantic segmentation of tv news broadcast videos," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 6191–6225, 2020.

[13] A.-G. Chifu and S. Fournier, "Segchain: Towards a generic automatic video segmentation framework, based on lexical chains of audio transcriptions," in *Proceedings of the 6th international conference on web intelligence, mining and semantics*, 2016, pp. 1–8.

[14] S. Iwata, W. Ohyama, T. Wakabayashi, and F. Kimura, "Recognition and connection of moving captions in arabic tv news," in *2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR).* IEEE, 2017, pp. 163–167.

[15] M. Pala, L. Parayitam, and V. Appala, "Real-time transcription, keyword spotting, archival and retrieval for telugu TV news using ASR," *International Journal of Speech Technology*, vol. 22, no. 2, pp. 433–439, 2019.

[16] R. Bansal and S. Chakraborty, "Visual Content Based Video Retrieval on Natural Language Queries," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. New York, NY, USA: ACM, 2019, pp. 212–219.

[17] Z. Dong and X. Lv, "Subject extraction method of urban complaint data," in *Proceedings of the IEEE International Conference on Big Knowledge (ICBK)*, 2017, pp. 179–182.

[18] B. Mocanu, R. Tapu, and T. Zaharia, "Automatic extraction of story units from TV news," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2017, pp. 414–415.

[19] T. Zlitni, B. Bouaziz, and W. Mahdi, "Automatic topics segmentation for TV news video using prior knowledge," *Multimedia Tools and Applications*, vol. 75, no. 10, pp. 5645–5672, 2016.

[20] Z. Liu and Y. Wang, "TV News Story Segmentation Using Deep Neural Network," in *Proceedings of the IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2018, pp. 1–4.

[21] T. R. Gruber *et al.*, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–221, 1993.

[22] M. Gruninger, "Ontology: applications and design," *Commun. ACM*, vol. 45, no. 2, 2002.

[23] C. Roussey, F. Pinet, M. A. Kang, and O. Corcho, "An introduction to ontologies and ontology engineering," in *Ontologies in Urban development projects*. Springer, 2011, pp. 9–38.

[24] I. Peters, *Folksonomies. Indexing and retrieval in Web 2.0*. Walter de Gruyter, 2009.

[25] X. Yang, R. Gao, Z. Han, and X. Sui, "Ontology-based hazard information extraction from chinese food complaint documents," in *International Conference in Swarm Intelligence*. Springer, 2012, pp. 155–163.

[26] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," 2010.

[27] O. Corcho, "Ontology based document annotation: trends and open research problems," *International Journal of Metadata, Semantics and Ontologies*, vol. 1, no. 1, pp. 47–57, 2006.

[28] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle, "Ontology-based extraction and structuring of information from data-rich unstructured documents," in *Proceedings of the seventh international conference on Information and knowledge management*, 1998, pp. 52–59.

[29] D. M. Blei, "Introduction to probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2011.

[30] A. Lesnikowski, E. Belfer, E. Rodman, J. Smith, R. Biesbroek, J. D. Wilkerson, J. D. Ford, and L. Berrang-Ford, "Frontiers in data analytics for adaptation research: Topic modeling," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 10, no. 3, p. e576, 2019.

[31] M. G. d. Oliveira, "Ontology-driven urban issues identification from social media." Ph.D. dissertation, Federal University of Campina Grande (UFCG), 2016.

[32] M. Jarrar, "Towards effectiveness and transparency in e-business transactions: an ontology for customer complaint management," in *Semantic Web for Business: Cases and Applications*. IGI Global, 2009, pp. 127–149.

[33] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.

[34] C. Stadler, J. Lehmann, K. Höffner, and S. Auer, "Linkedgeodata: A core for a web of spatial open data," *Semantic Web*, vol. 3, no. 4, pp. 333–354, 2012.

[35] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[36] S. Cox, C. Little, J. Hobbs, and F. Pan, "Time ontology in owl," *W3C recommendation*, vol. 19, 2017.

[37] Selenium, "Selenium Library." [Online]. Available: https://www.selenium.dev/ [accessed: 2020-03-02]

[38] R. Gonzalez et al., "YouTubeDL." [Online]. Available: https://github.com/ytdl-org/youtube-dl [accessed: 2020-03-02]

[39] A. Zhang, "Selenium." [Online]. Available: https://github.com/Uberi/speechrecognition [accessed: 2020-03-02]

[40] Explosion AI, "Spacy." [Online]. Available: https://spacy.io/ [accessed: 2020-03-02]

[41] NLTK Project, "The Natural Language Toolkit." [Online]. Available: https://radimrehurek.com/gensim/ [accessed: 2020-03-02]

[42] D. Carriere et al., "Geocoder." [Online]. Available: https://geocoder.readthedocs.io/ [accessed: 2020-03-02]

[43] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora." [Online]. Available: https://radimrehurek.com/gensim/ [accessed: 2020-03-02]

[44] G1 Paraiba, "JPB1 TV News program official website." [Online]. Available: http://g1.globo.com/pb/paraiba/jpb-1edicao/videos/ [accessed: 2020-03-02]

[45] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Comput. Surv.*, vol. 49, no. 2, pp. 31:1–31:50, 2016.

[46] D. Fury, "TrueCase." [Online]. Available: https://github.com/daltonfury42/truecase [accessed: 2020-03-02]

[47] S. Han, "Googletrans." [Online]. Available: https://github.com/ssut/py-googletrans [accessed: 2020-03-02]

[48] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 2016, pp. 785–794.

[49] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602 – 610, 2005, iJCNN 2005.

Context and Georeferencing of Archaeological, Prehistorical, and Geoscientific Object Entities, Resulting from Integration of Knowledge Resources' Development and Universal Conceptual Knowledge Pattern Matching

Claus-Peter Rückemann
Westfälische Wilhelms-Universität Münster (WWU), Germany;
Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF), Germany;
Leibniz Universität Hannover, Germany
Email: ruckema@uni-muenster.de

Abstract—**This paper presents the long-term research results on context creation and georeferencing, deploying advanced knowledge based mining, enabled by conceptual knowledge frameworks. The goal of this fundamental research is to systematically develop advanced information science tools and knowledge-based methods, gaining new insights for complex archaeological, prehistorical, and geoscientific contexts. The vast resources and data collections are otherwise not practically available to archaeologists and geoscientists. The paper presents the methodological base of an innovative algorithm framework of conceptual knowledge pattern matching, allowing the consideration of complementary and descriptive knowledge of meaning and intrinsic object properties. The research is illustrated by practical implementations of knowledge pattern matching, including processing and developing multi-disciplinary and multi-lingual knowledge object entities and resources. This specialised research concentrates on geoscientific context and georeferencing. The demonstration of a previously unpublished practical information science case study from classical archaeology, prehistory, and geosciences shows automated knowledge context creation and georeferencing.**

***Keywords*–Knowledge Context Creation; Georeferencing; Archaeology and Geosciences; Superordinate Knowledge; Conceptual Knowledge Pattern Matching Methodology.**

I. Introduction

Advanced creation of context and georeferencing from multi-disciplinary archaeological, prehistorical, and geoscientific object entities is a most challenging task. So far, there is no consistent, practical concept available relieving the fundamental deficits of respective resources, especially motivating new fundamental ways of concepts' integration.

This extended research is based on the creation of advanced knowledge resources and implementation of the Conceptual Knowledge Pattern Matching (CKPM), which was presented at the GEOProcessing 2020 conference in Valencia, Spain [1].

This paper presents developments, which are complementary to conceptual knowledge deployed in knowledge resources and processing method used so far. The new developments closely integrate with features enabled by knowledge-centric development of resources and objects. The presented cases focus on basic phonetic associations and topographic and cartographic context creation and visualisation, starting with advanced knowledge mining.

Knowledge Mining is supported by a number of common methods and algorithms, e.g., string pattern matching algorithms, associative, comparative, and phonetic algorithms. All these achievements deal with distinct extrinsic properties of respective entities in very limited ways.

The motivation for creating the CKPM was the lack of suitable facilities for an advanced matching of 'meaning' when implementing mining solutions in context of complex multi-disciplinary and multi-lingual Knowledge Resources. Especially, conceptual knowledge can relate to any of factual, conceptual, and procedural knowledge. To a comparable extent, metacognitive knowledge can also relate to any of factual, conceptual, and procedural knowledge but the formalisation of conceptual knowledge is different in a way that universal implementations can be deployed in flexible and still consistent ways. A practical approach for knowledge pattern matching is the fundament for this research. Its methodological fundaments and the practical integration with the development of universal knowledge resources will be presented in the following sections, targetting on the practical creation of object entities' context and georeferencing.

The rest of this paper is organised as follows. Section II provides the background of knowledge complements and formalisation of meaning. Section III introduces the previous work and components. Sections IV presents methodology, method, and implementation. Sections V, VI, and VII present the implementation, matching process, and resulting tables. Section VIII provides an implementation of module integration for the result of context creation and georeferencing. Sections IX and X discuss the main achievements and summarise conclusions and future work.

II. From Meaning and Structures to Knowledge Complements

Knowledge, meaning, and patterns form relations, which may require some introduction.

The concept of meaning differs from the concept of signification. Semantic and syntactic structures do not suffice to determine the discursive meaning of an expression [2]. Discourse means a way of speaking. On the one hand, grammatically correct phrases may lack discursive meaning. On the other hand, grammatically incorrect sentences may be discursively meaningful.

Such formalised approaches require a solid understanding of interpretation [3] and application of methods. Fundamental approaches, especially, the semiotic triangle of symbol, thought/Reference, and referent [4] are fundamental for understanding principles.

Knowledge and meaning are closely tied with intrinsic and extrinsic properties. Therefore, understanding of intrinsic and extrinsic properties of entities is significant for any context.

This is nevertheless true for any case of natural language, esp. considering langage, langue, and parole [5].

Creating practical approaches requires algorithms. An algorithm is a process or set of rules to be followed in problem-solving operations. In general, algorithms cannot, by their fundamental nature, handle intrinsic and extrinsic properties to the same quality and extent. For example, an intrinsic property of a word object is the meaning in mind, the 'lemma'. An extrinsic property of a word object can be a written word. Extrinsic properties do not reflect meaning and insight as their representations do not generally allow reasonable results. Best practice provides us with solid, complementary knowledge concepts and methodologies allowing to create advanced methods.

Data do not have or carry meaning. Therefore, understanding meaning is of major significance in information science when dealing with improving formalisation processes and creating 'logos based' analogies along with cognitive processes. Commonly, cognition (cognitio, from Latin cognoscere, "get to know") is the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses (Source: Oxford dictionary). Analogy (from Greek analogia, ἀναλογία, "proportion") is a cognitive process of transferring information or 'meaning' from a particular subject, the analogue or source, to another, the target.

Nevertheless, aspects of meaning can be described using knowledge complements, e.g., considering factual, conceptual, procedural, and metacognitive knowledge [6]. Especially, conceptual knowledge can relate to any of factual, conceptual, and procedural knowledge. To a comparable extent, metacognitive knowledge can relate to any of factual, conceptual, and procedural knowledge. A practical approach for knowledge pattern matching will be presented in the following sections.

III. Previous work, components, and resources

The fundaments of terminology and understanding knowledge are layed out by Aristotle being an essential part of 'Ethics' [7]. Information sciences can very much benefit from Aristotle's fundaments and a knowledge-centric approach [6] but for building holistic and sustainable solutions, supporting a modern definition of knowledge [8], they need to go beyond the available technology-based approaches and hypothesis [9] as analysed in Platon's Phaidon.

Making a distinction and creating interfaces between methods and the implementation applications, the results of this research are illustrated here along with the practical example of the Knowledge Mapping methodology [10] enabling the creation of new object and entity context environments, e.g., implementing methods for knowledge mining context.

The means to achieve such recommendations even for complex scenarios is to use the principles of Superordinate Knowledge, integrating arbitrary knowledge. The core assembly elements of Superordinate Knowledge [11] are:

- Methodology.
- Implementation.
- Realisation.

Separation and integration of assemblies have proven beneficial for building solutions with different disciplines and different levels of expertise. Comprehensive focussed subsets of conceptual knowledge can also provide excellent modular and standardised complements for information systems component implementations, e.g., for environmental information management and computation [12].

For the implementation of case studies, the modules are built by support of a number of major components and resources, which can be used for a wide range of applications, e.g., creation of resources and extraction of entities. The Universal Decimal Classification (UDC) [13] is the world's foremost document indexing language in the form of a multi-lingual classification scheme covering all fields of knowledge and constitutes a sophisticated indexing and retrieval tool. The UDC is designed for subject description and indexing of content of information resources irrespective of the carrier, form, format, and language. UDC is an analytico-synthetic and facetted classification. It uses a knowledge presentation based on disciplines, with synthetic features. UDC schedules are organised as a coherent system of knowledge with associative relationships and references between concepts and related fields. The UDC allows an efficient and effective processing of knowledge data and provides facilities to obtain a universal and systematical view on classified objects. UDC-based references in this publication are taken from the multi-lingual UDC summary [13] released by the UDC Consortium [14] under a Creative Commons license [15]. Facets can be created with any auxiliary tables, e.g., auxiliaries of place and space, time, language, and form as well as general characteristics, e.g., properties, materials, relations, processes, and operations, persons and personal characteristics. Module examples are employing Perl Compatible Regular Expressions (PCRE) [16] syntax for specifying common string patterns and Perl [17] for component wrapping purposes with this information science case study in archaeology, prehistory, and geosciences.

In addition, context data, visualisation modules, and phonetic modules were developed and integrated for benefit of this research. The used components will be referred to in the respective following sections.

IV. Methodology and implementation

The implementation strictly follows the fundamental methodological algorithm base.

A. Methodological algorithm base

The Conceptual Knowledge Pattern Matching (CKPM) methodology targets providing and accessing knowledge object patterns. This methodological algorithm framework is based on the Superordinate Knowledge Methodology, which allows systematical use and thorough processing by the steps:

1) Selecting knowledge objects.
2) Accessing knowledge object patterns.
3) Thorough processing of object entities and references.
4) Object entity analysis, knowledge complements' based.
5) Result formation.

The respective accessing includes the organisation and structures used with the objects and entities. Object patterns need to be accessible to an extent and quality, which allows a sufficient processing for the respective scenario. The requirements for specific scenarios will therefore be individual. The processing includes making use of the characteristics and features of the respective implementations of the knowledge based frameworks providing a conceptual base for a certain method. The conceptual knowledge complements referred from knowledge objects can have their origins from manual as well as from automated processes. For the implementation and realisation, the framework providing the base conceptual knowledge reference patterns is the UDC. The results in this publication use the UDC Summary Linked Data (Main Tables, [18]). Creating facets and patterns can also make use of the common auxiliary signs being part of the UDC framework [19]. The following advanced employment of conceptual knowledge (UDC) is far beyond common application of universal classification.

B. Implemented method

An implementation of a CKPM based method requires accessible objects and a suitable conceptual framework base for processing and automation. The methodic implementation illustrated here enables to employ an UDC framework appropriate for systematical use, implemented by the steps:

1) Knowledge Resources' objects.
2) Accessing formalised conceptual knowledge object pattern description based on UDC, e.g., including geoscientific context and georeferencing.
3) Processing procedure via pipelines, employing UDC knowledge and forks.
4) Entity analysis, based on UDC framework references.
5) Result formation on base of Knowledge Resources' objects, retaining conceptual knowledge.

In this case, meaning is described by conceptual patterns, which can be searched and analysed. Processing algorithms can follow the given organisation, e.g., the decimal organisation of the UDC, following available forks as will be illustrated for the matching process in the following sections. Processing and analysis includes primary, decimal conceptual knowledge and associated multi-dimensional knowledge in context of the object entities. The method allows advanced data-centric computing procedures. In practice, the facility for consistently describing knowledge is a valuable quality, esp., conceptual knowledge, e.g., using the UDC and its editions.

C. Implemented conceptual knowledge framework and target

Targeting practical use for advanced geoscientific information and expert systems, conceptual geoscientific and geographic mapping and referencing are required. Geographic conceptual knowledge pattern entities are created based on UDC code references [20] of geography, biography, history. Table I shows an implementation excerpt.

TABLE I. CONCEPTUAL KNOWLEDGE PATTERN MATCHING: IMPL. UDC REFERENCES OF GEOGRAPHY, BIOGRAPHY, HISTORY (EXCERPT).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:902	Archaeology
UDC:903	Prehistory. Prehistoric remains, artefacts, antiquities
UDC:904	Cultural remains of historical times
UDC:908	Area studies. Study of a locality
UDC:91	Geography. Exploration of the Earth and of individual countries. Travel. Regional geography
UDC:912	Nonliterary, nontextual representations of a region
UDC:92	Biographical studies. Genealogy. Heraldry. Flags
UDC:93/94	History
UDC:94	General history

A geoscientific / archaeology example from the case studies and implementations for geoscientific information systems and application components is used for illustration in the next sections. The example will show a tiny subset of the comprehensive, universal conceptual knowledge used.

The above conceptual knowledge contains all the references for geographic context, which includes the conceptual knowledge regarding geographic data, e.g., geoinformation and geodescriptive knowledge. The relevant conceptual knowledge required for geoscientific context is provided by references from natural sciences' context. Any of the conceptual knowledge can be used in any stage of a CKPM process, e.g., in start, intermediate, and target specifications.

Natural sciences related conceptual knowledge pattern entities are created based on UDC code references [21] of mathematics and natural sciences. An excerpt of the implementation is shown in Table II.

TABLE II. CONCEPTUAL KNOWLEDGE PATTERN MATCHING: IMPL. UDC REFERENCES OF MATHEMATICS AND NATURAL SCIENCES (EXCERPT).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:51	Mathematics
UDC:52	Astronomy. Astrophysics. Space research. Geodesy
UDC:53	Physics
UDC:54	Chemistry. Crystallography. Mineralogy
UDC:55	Earth Sciences. Geological sciences
UDC:550.3	Geophysics
UDC:551	General geology. Meteorology. Climatology. Historical geology. Stratigraphy. Palaeogeography
UDC:551.21	Vulcanicity. Vulcanism. Volcanoes. Eruptive phenomena. Eruptions
UDC:551.7	Historical geology. Stratigraphy. Palaeogeography
UDC:551.8	Palaeogeography
UDC:551.24	Geotectonics
UDC:56	Palaeontology
UDC:57	Biological sciences in general
UDC:58	Botany
UDC:59	Zoology

Time related conceptual knowledge pattern entities are created based on UDC code references [22], especially the auxiliaries of time). Table III shows an implementation excerpt.

TABLE III. CONCEPTUAL KNOWLEDGE PATTERN MATCHING: IMPLEMENTED UDC REFERENCES, AUXILIARIES OF TIME (EXCERPT).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:"0"	First millennium CE
UDC:"1"	Second millennium CE
UDC:"2"	Third millennium CE
UDC:"3/7"	Time divisions other than dates in Christian (Gregorian) reckoning
UDC:"3"	Conventional time divisions and subdivisions: numbered, named, etc.
UDC:"4"	Duration. Time-span. Period. Term. Ages and age-groups
UDC:"5"	Periodicity. Frequency. Recurrence at specified intervals.
UDC:"6"	Geological, archaeological and cultural time divisions
UDC:"61/62"	Geological time division
UDC:"63"	Archaeological, prehistoric, protohistoric periods and ages
UDC:"7"	Phenomena in time. Phenomenology of time

Spatial conceptual knowledge pattern entities are created based on UDC code references [23], especially the auxiliaries of spatial features and place (UDC (1/9)), (Table IV).

TABLE IV. CONCEPTUAL KNOWLEDGE PATTERN MATCHING: IMPL. UDC REFERENCES, AUXILIARIES OF SPATIAL FEATURES AND PLACE (EXCERPT).

Code / Sign Ref.	*Verbal Description (EN)*
UDC:(1)	Place and space in general. Localization. Orientation
UDC:(2)	Physiographic designation
UDC:(3)	Places of the ancient and mediaeval world
UDC:(31)	Ancient China and Japan
UDC:(32)	Ancient Egypt
UDC:(33)	Ancient Roman Province of Judaea. The Holy Land. Region of the Israelites
UDC:(34)	Ancient India
UDC:(35)	Medo-Persia
UDC:(36)	Regions of the so-called barbarians
UDC:(37)	Italia. Ancient Rome and Italy
UDC:(38)	Ancient Greece
UDC:(399)	Other regions. Ancient geographical divisions other than those of classical antiquity
UDC:(4)	Europe
UDC:(5)	Asia
UDC:(6)	Africa
UDC:(7)	North and Central America
UDC:(8)	South America
UDC:(9)	States and regions of the South Pacific and Australia. Arctic. Antarctic

Knowledge Resources' objects carry respective conceptual UDC facets and references, including georeferences.

V. BASIC PRINCIPLE PROCESSING IMPLEMENTATION

Regarding an implementation ('`lxgrep`'), a basic routine preparing object entity input into a common structure is illustrated in Figure 1.

```
if (/^(\S)(.*)/./^              /||/^$/||/^ *$/) {
  s/^(\S.*)\n/\@ENTRY\@$1@@/;
  s/^(.*)\n/\1@@/;
  s/\@ENTRY\@/\n/;
  open(TMPFILE,">>$tempfile"); print TMPFILE "$_"; close(
   TMPFILE);
}
```

Figure 1. Basic routine preparing input entries (excerpt).

An associated elementary system call implementing a basic regular search is shown in Figure 2.

```
system("egrep␣-h␣$temppat␣$ARGV[0].tmp␣>␣$ARGV[0].grep.
 tmp");
system("mv␣$ARGV[0].grep.tmp␣$ARGV[0].tmp");
```

Figure 2. Elementary system call for a basic regular search (excerpt).

An element for a simple system sort based function used with the above search is shown in Figure 3.

```
print "\tsorting␣entries...\n";
system("sort␣-f␣-k␣1,14␣<$tempfile>$tempfile.out");
unlink $tempfile;
```

Figure 3. Element of simple system call sort function (excerpt).

A simple backformatting routine is given in Figure 4.

```
print "\tbackformatting␣entries...\n";
system("perl␣-e␣'while␣(<>){s/@@/\n/g;chop;print␣$_}'␣<
 $ARGV[0].tmp>$ARGV[0].sort");
unlink "$ARGV[0].tmp";
```

Figure 4. Simple backformatting routine (excerpt).

For further structural, technical details, and pipelining please see the references for the case studies given in the text.

VI. NEW MATCHING PROCESS AND PROCESSING

The new framework of the matching process and processing includes following the conceptual knowledge forks. Here, the primary, decimal reference forks of the UDC are used for implementation, which provide the red line forks within universal knowledge. Matching is beyond non-conceptual knowledge, e.g., natural language processing and string pattern matching. Especially, country and border concepts cannot be used for specification, e.g., ancient and modern border lines fail to be useful. The process enables places in ancient Greece and Rome, from archaeological and prehistoric times associated with places in the ancient and modern world to be described, e.g., references of the type `UDC:..."63"(37)` and `UDC:..."63"(38)`. Trigger question can be 'Can archaeological artefacts' objects of a certain context be associated with earth science objects?'. A symbolic writing specifying a conceptual expression is shown in Figure 5.

```
STRT:[UDC:.*?90]
CTXT:[[UDC:.*?\(.*?38.*?\)]|[UDC:.*?\"6.*?\"]].*[[UDC
 :.*?\"6.*?\"]|[UDC:.*?\(.*?38.*?\)]]
SRCH:[[UDC:.*?55]|[UDC:.*?912]]
```

Figure 5. Example for symbolic writing of pattern expression (excerpt).

A systematic concept of conceptual knowledge implementation allows advanced features, e.g., pattern range variations, pattern permutations. A basic serial pipeline implementation example test for knowledge objects in `<input>` is shown in Figure 6.

```
cat <input> | lxgrep "'%%IML:.*?UDC:.*?\(38.*?\)'" |
 lxgrep "'%%IML:.*?UDC:.*?\"6.*?\"'" <outputctxt>
cat <outputctxt> | lxgrep "'%%IML:.*?UDC:.*?90'"
cat <outputctxt> | lxgrep "'%%IML:.*?UDC:.*?55'" | lxgrep
  "'%%IML:.*?UDC:.*?912'" | lxgrep LATLON:
```

Figure 6. Example for serial pipeline implementation (excerpt).

The pipeline includes objects containing and referring to latitude / longitude objects. The trackable spatial/place related fork process within the conceptual pattern entity group is illustrated in Figure 7.

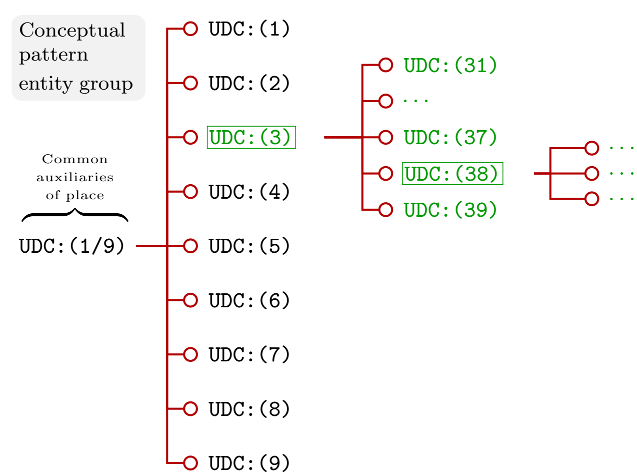


Figure 7. Matching process: Primary, decimal (UDC) conceptual knowledge forks, auxiliaries of spatial features and place (excerpt).

The processing successfully follows the "Ancient Greece" fork. Figure 8 illustrates the fork process within the conceptual pattern entity group for the related conceptual knowledge regarding time.

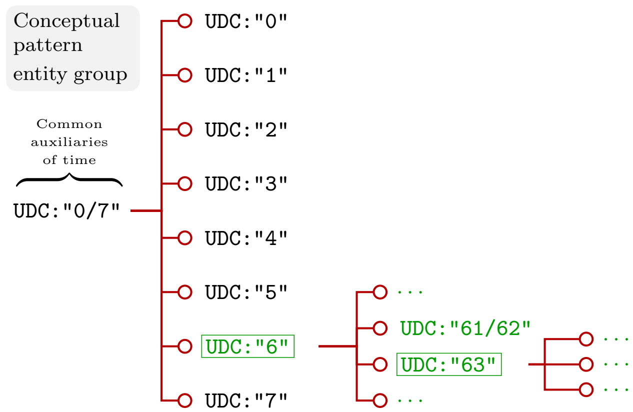


Figure 8. Matching process: Primary, decimal (UDC) conceptual knowledge forks, auxiliaries of time (excerpt).

The processing successfully follows the "geographical/historical" and "natural sciences" fork. The main tables of the conceptual knowledge are managed in the same way within the respective conceptual pattern entity groups (Figure 9).

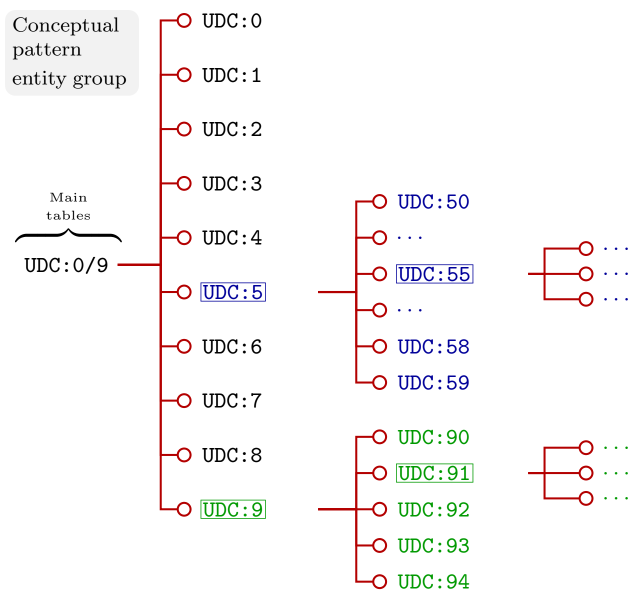


Figure 9. Matching process: Primary, decimal (UDC) conceptual knowledge forks, main tables, including earth sciences and geography (excerpt).

The processing successfully follows the "Earth sciences" and "Geography" forks. These procedures referencing to a formalised [24], practical framework of conceptual knowledge embrace all the relevant universal knowledge, e.g., including natural sciences and geosciences, archaeology, philosophy, and history. The results of removing in the domain of knowledge and removing in the domain of mathematics are not the same. In principle, abstraction means removing [25]. In the mathematical domain, removing is mostly formalised by subtraction [26].

In general, any universal conceptual knowledge framework can be used, which enables a systematical processing and which is universal and consistent. Nevertheless, UDC has proven to be by far the most universal, practical, and efficient choice for multi-disciplinary and multi-lingual context. Therefore, UDC references are used for this publication, illustration, and the practical implementations for this research.

Demonstrated meaning patterns can be used plain, e.g., on conceptual fundament, or integrating references between objects containing non-conceptual knowledge, too. The development target will certainly depend on intended case scenarios and capacities of resources' and module developers. This research can only demonstrate some principle, major features of UDC in context of an implementation done for the given practical scenario. However, it is always possible to parametrise conceptual knowledge depth and width of matches for any case due to the structure of the UDC implementation.

VII. Resulting Match Tables

Following the above archaeology-geosciences case of matching process and processing, the resulting match tables contain the references to conceptional and associated multi-dimensional knowledge in context of the object entities. The resulting start match table of object entities (Table V) contains entities and references on details of mythological and archaeological context.

Table V. Resulting Conceptual Knowledge Pattern Matching intermediate start ('UDC:90') match table (excerpt).

Object Entity	*Reference Data (excerpt)*
Poseidon	DESC MYTH SYN LOC UDC … CITE:[27], [28], [29], [30]
Polybotes/-is	DESC MYTH SYN LOC UDC … CITE:[27], [29]
Polyvotes/-is	DESC MYTH SYN LOC UDC … CITE:[27], [29] (transcr.)

These entities contain descriptions, including transcriptions, transliterations, translations, mythology references, synonyms, location references, UDC references, and citation sources. The citations refer to respective associations of the figured programme with Poseidon and the giant Polybotes / Polybotis / Polyvotes / Polyvotis and further references to the details of mythological context of realia objects, respectively to Parthenon metopes (Acropolis, Athens).

The result match table of object entities (Table VI) contains entities and references on details of natural sciences context and georeferences.

Table VI. Resulting Conceptual Knowledge Pattern Matching intermediate result ('UDC:55') match table (excerpt).

Object Entity	*Reference Data (excerpt)*
Kos	DESC VOLC VNUM GRC LATLON UDC …
Methana	DESC VOLC VNUM GRC LATLON UDC …
Milos	DESC VOLC VNUM GRC LATLON UDC …
Nisyros	DESC VOLC VNUM GRC LATLON UDC …
Santorini	DESC VOLC VNUM GRC LATLON UDC …
Yali	DESC VOLC VNUM GRC LATLON UDC …

The entities in the respective match tables contain descriptions, volcanological references, volcano numbers, country references, latitude and longitude location references, UDC references, and further references. A resulting object is shown in Figure 10. Its respective media object entities refer to archaeology associated with Poseidon and Polyvotis.

```
Nisyros              [Volcanology, Geology, Archaeology]:
                     Volcano, Type: Strato volcano, Island,
                     Country: Greece, Subregion Name: Dodecanese Islands,
                     Status: Historical, Summit Elevation: 698\UD{m}. ...
                     Craters: ..., VNUM: 0102-05=, ...
                     %%IML: UDC:[911.2+55],[930.85],[902]"63"(4+38+23+24)=14
                     %%IML: UDC:[912] ...
                     %%IML: media:...{UDC:[911.2+55],"63"(4+38+23)}...jpg
                     Stefanos Crater, Nisyros, Greece.
                     LATLON: 36.578345,27.1680696
                     %%IML: GoogleMapsLocation: https://www.google.com/...@36
                      .578345,27.1680696,337m/...
                     Little Polyvotis Crater, Nisyros, Greece.
                     LATLON: 36.5834105,27.1660736 ...
```

Figure 10. Result object entity from Knowledge Resources: Nisyros object, Greece, containing media object entities and georeferences (excerpt).

As requested, the object contains / refers to latitude / longitude and conceptual knowledge, together with factual knowledge and media references.

Figure 11 shows media object entities based on the conceptual knowledge pattern matching process, an object entity at process start (Figure 11(a)), from archaeological artefacts, and a resulting reference object (Figure 11(b)), from natural objects. The media object entities and their context represent the result of the requested knowledge pattern matching, including respective georeferencing properties.

VIII. Integration modules for result context creation and georeferencing

Commonly, it is a challenging question how the integration of multi-disciplinary knowledge object instances and context can be achieved for further deployment.

A. Phonetic modules and values

The deployed implementation modules support a 'silken' selection [31], phonetic modules, which can be specified and closely integrated with conceptual context.

The knowledge resources can be used by any algorithm suitable for a defined workflow. One of the available module implementing a silken selection based on the Soundex principle is the `knowledge_sndx_standard` application. The historical Soundex [32] is a phonetic algorithm for indexing names by sound. The goal with this algorithm is to encode homophones so that they can be represented by the same resulting code in order to match persons' name despite differences in writing and spelling [33]. The basic algorithm mainly encodes consonants. Vowels are not encoded unless being a first letter. The U.S. Government is using a modified modern rule set [34] for purposes in census and archives. The original intention was to catch the English pronunciation, anyhow there are many different implementations in use today.

The selection module, modelled after the standard Perl implementation [35], for computing LX Soundex codes [36], can be adapted for different procedural context and programming concepts [37], [38]. The various workflows can define and integrate their own Soundex codes for different purposes and topics.

A large number of different modules and algorithms were developed for this research, even including various phonetic algorithms for respective content and context. Listing 11 shows a fully functional excerpt Perl source code module (`knowledge_sndx_grlat`) modelled after the standard Perl implementation and developed for the respective Greek and Latin context.

```perl
#!/usr/bin/perl
#
# knowledge_sndx_grlat -- (c) LX Project -- CPR
 1992, 2012, 2019
#

$string=$ARGV[0];
$sndx_nocode = undef;

sub knowledge_sndx_grlat
{
  local (@s, $f, $fc, $_) = @_;
  push @s, '' unless @s;

  foreach (@s)
  {
    $_ = uc $_;
    tr/A-Z//cd;

    if ($_ eq '')
    {
      $_ = $sndx_nocode;
    }
    else
    {
      ($f) = /^(.)/;
      tr/-AEHIOUWYBFPVCGJKQSXZDTLMNR
       /00000000011112202222234556/;
      ($fc) = /^(.)/;
      s/^$fc+//;
      tr///cs;
      tr/0//d;
      $_ = $f . $f . $_ . '000';
      s/^(.{5}).*/$1/;
      $_=~s/BB/BV/;
      $_=~s/VV/BV/;
    }
  }

  wantarray ? @s : shift @s;
}

$code = knowledge_sndx_grlat $string;
print ("SNDX-grlat:$code:$string\n");

##EOF:
```

Figure 11. LX Soundex SNDX-grlat module Perl source code (excerpt), considering language dependent homophonic partial word transcriptions.

Table VII shows the result of the selection module for respective transcriptions in this information science case study.

Table VII. Phonetic Soundex SNDX-grlat codes for transcripts of Polybotes and homophonic partial words (excerpt).

Object Entity	*Soundex Code (excerpt)*
Polybotes	SNDX-grlat:PP413:Polybotes
Polybotis	SNDX-grlat:PP413:Polybotis
Polyvotes	SNDX-grlat:PP413:Polyvotes
Polyvotis	SNDX-grlat:PP413:Polyvotis
-botes	SNDX-grlat:BV320:-botes
-botis	SNDX-grlat:BV320:-botis
-votes	SNDX-grlat:BV320:-votes
-votis	SNDX-grlat:BV320:-votis

The special phonetic code algorithm can consider possible transcriptions in respective languages for code results, which reflect a well comparable phonetic situation even for different transcriptions of the same entities.

(a) Metope, New Acropolis Museum, Athens, (CPR, DIMF, 2019).

(b) Volcano crater, island of Nisyros, Dodecanese Islands, Greece, (CPR, DIMF, 2019).

Figure 11. Result based on the conceptual knowledge pattern matching process, via intermediate match table (Table VI): (a) an artefact, metope (EAST VI), Parthenon, (Archaeology Digital Object Archive, 2019), and a resulting georeferenced object, (b) a natural object (Geosciences Digital Object Archive, 2019).

B. Knowledge reference tree: Phonetic and conceptual

Associated relations of object entities may span more than a distinctive conceptual knowledge reference group. Table VIII shows an excerpt of object entities and associated sources and reference data. The excerpt reflect presented the information science case study. Reference data excerpts list UDC code references and Soundex code references (SNDX-standard) for the named object entities.

Colour codes of the generated knowledge context reflect the results of the knowledge pattern matching, especially the result group (yellow) and result group contexts (green/blue).

TABLE VIII. KNOWLEDGE REFERENCE TABLE: UDC CODE REFERENCES AND PHONETIC (SOUNDEX, SNDX-GRLAT) ASSOCIATION (EXCERPT).

Object Entity and Sources	*KR*	*Reference Data (excerpts)*	
		UDC Code	*Soundex Code*
Parthenon, Acropolis, Athens	O	`UDC:902/904...`	`SNDX:PP635`
Polybotes, [27], [29] (transcr.)	O	`UDC:902/904...`	`SNDX:PP413`
			`SNDX:BV320`
Poseidon, [27], [28], [29], [30]	O	`UDC:902/904...`	`SNDX:PP235`
...			
Kos (Volcano)	C	`UDC:550.3...`	`SNDX:KK200`
Methana (Volcano)	C	`UDC:550.3...`	`SNDX:MM350`
Milos (Volcano)	C	`UDC:550.3...`	`SNDX:MM420`
Nisyros (Volcano)	CO	`UDC:550.3...`	`SNDX:NN262`
Polyvotis (Volc. Crater)	CO	`UDC:550.3...`	`SNDX:PP413`
			`SNDX:BV320`
Stefanos (Volc. Crater)	CO	`UDC:550.3...`	`SNDX:SS315`
Santorini (Volcano)	C	`UDC:550.3...`	`SNDX:SS536`
Yali (Volcano)	C	`UDC:550.3...`	`SNDX:YY400`
...			

Knowledge Resources (KR) can contain objects and entities of different organisation and structure, especially, collection (O) object entities and container (C) object entities. Reference data, especially UDC codes and phonetic Soundex codes, can be deployed for complex scenarios, which benefit from advanced referenced besides knowledge content and context of object entities and referenced sources.

There are two categories of object entities regarding a georeferencing context: Object entities, which cannot be logically attributed with a georeferencing and those, which have properties allowing georeferencing attributation. The first category (upper group) in this case is represented by a `UDC:902/904` context group, whereas the second category (lower group) is represented by a `UDC:550.3` context group.

The groups are linked by conceptual knowledge as some entities refer to both special collections and containers (CO). An additional, significant reference link between the group calculates from the phonetic `SNDX:PP413`/`SNDX:BV320` references.

C. Reference to automated context and georeferencing

Those associated object entities of different groups (Table VIII), which carry conceptual references and georeferencing can be selected by available properties and criteria, especially conceptual knowledge and phonetic means.

Table IX shows an excerpted result of the phonetic/conceptual knowledge reference tree process, the automatically associated georeferenced object entities.

TABLE IX. RESULT OF THE PHONETIC/CONCEPTUAL KNOWLEDGE REFERENCE TREE: GEOREFERENCED OBJECT ENTITIES (EXCERPT).

Object Entity	*Reference Data (excerpt)*	
	LATLON	
Kos (Volcano)	36.829	27.257
Methana (Volcano)	37.615	23.336
Milos (Volcano)	36.699	24.439
Nisyros (Volcano)	36.580	27.180
Parthenon, Acropolis, Athens	37.972	23.727
Polyvotis (Volc. Crater)	36.583	27.177
Santorini (Volcano)	36.404	25.396
Stefanos (Volc. Crater)	36.578	27.168
Yali (Volcano)	36.630	27.100

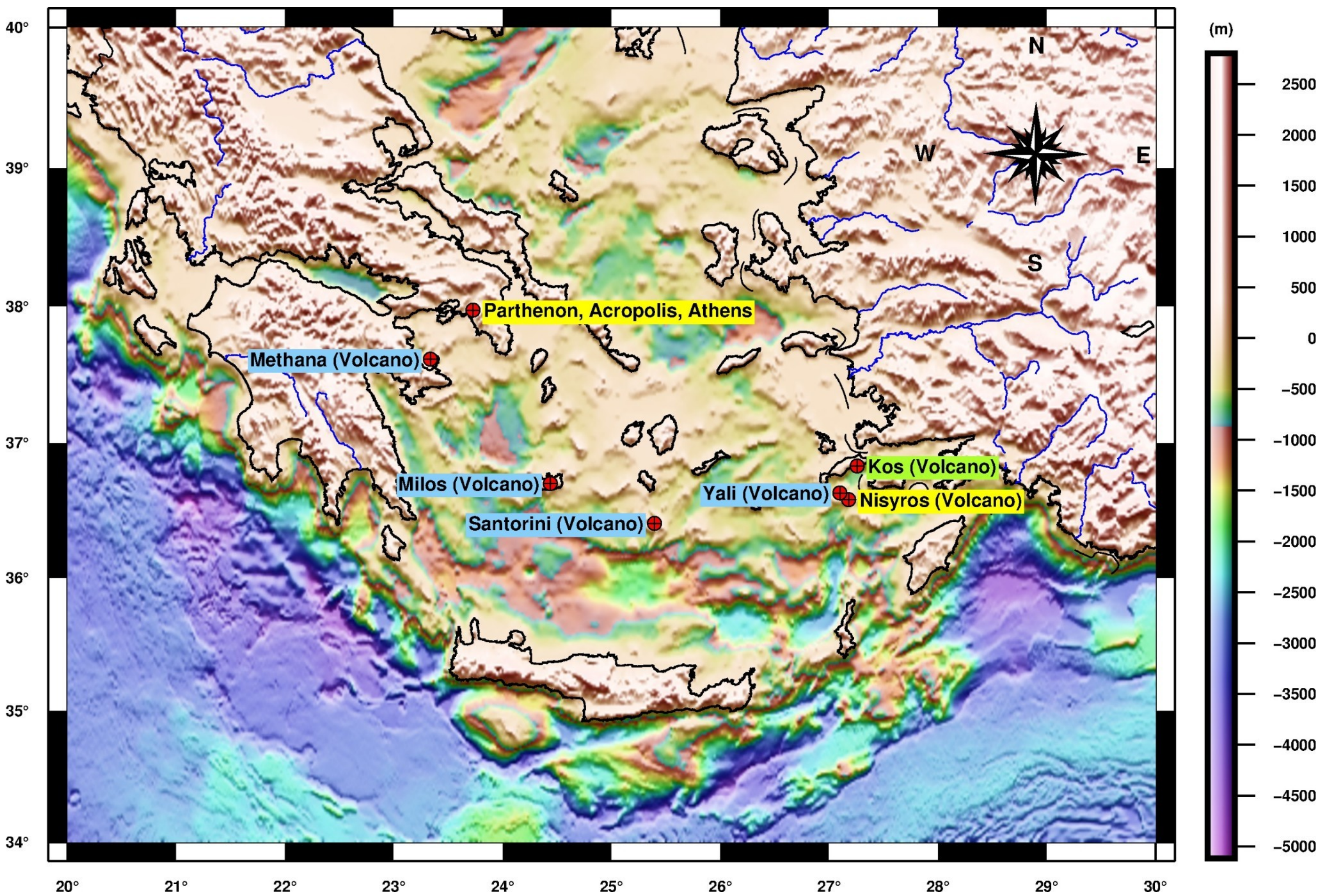


Figure 12. Automatically generated knowledge context (red bullet markers) of the conceptual knowledge pattern matching result group (yellow labels) and result group contexts (green/blue labels) in their topographic context, created based on georeferenced knowledge resources object entities and conceptual knowledge.

Consecutive context can be created by automatically generated mapping and visualisation modules, e.g., employing GMT routines, based on these results. Result context of the `UDC:902/904` context group are `UDC:550.3` context group object entities, which secondary context (Figure 12) is represented by respective geological and plate tectonics knowledge for the georeferenced area (Aegean Sea region). The new generated context integrates the various contexts and object entity representations in a multi-dimensional result. For these cases the module option is Mercator projection in order to provide a common base for comparisons of proper attributes and required flexibility.

Appropriate data was required for the topographic data related criteria. In the past, the georeferenced objects have been used with various data, e.g., with the Global Land One-kilometer Base Elevation Project (GLOBE) [39] and the 2-minute gridded global relief data (ETOPO2v2) [40].

For the required resolution of the results presented here, the knowledge resources had to be integrated with data based on the gridded ETOPO1 1 arc-minute global relief model data [41]. For special purposes data can be composed from various sources, e.g., adding Shuttle Radar Topography Mission (SRTM) data [42] from the Consultative Group on International Agricultural Research (CGIAR) [43].

The horizontal datum of ETOPO1 is World Geodetic System geographic, which was established in 1984 (WGS84) and later revised. The WGS84 specifications and references are provided by the National Geospatial-Intelligence Agency (NGA) [44] and as EPSG:4326 from the European Petroleum Survey Group Geodesy (EPSG) [45]. The vertical datum of ETOPO1 is "sea level". The source elevation data were not converted by the authors of ETOPO1 to a common vertical datum because of the large cell size of 1 arc-minute.

The Generic Mapping Tools (GMT) [46] suite application components are used for handling the spatial data, applying the related criteria, and for the visualisation.

IX. Discussion

The implemented method and the integration modules for result context creation and georeferencing has been a viable, efficient, and flexible solution in many case scenarios. Implementations are far not trivial but any discipline being able to ask questions as demonstrated should also be able to deploy the methodology and presented components for creating solid fundaments and practical solutions for challenging, complex scenarios, e.g., classification and dating of objects [47], geoscientific prospection, surveying [48], and knowledge [49], chorological and chronological context [50], can contribute to the fundaments of archeological and prehistorical cognition [51] and insight [52] regarding realia and abstract objects, knowledge, and contexts. The presented knowledge-based method and conceptual knowledge framework allow to address context very flexibly, e.g., in order to enable the metacognitive documentation of metacognitive and procedural knowledge of Geoscientific Information Systems or Geographic Information System analysis [53], filtering contextualised artistic representations [54] and managing object collections [55]. Knowledge-based approaches can also be beneficial without advanced knowledge resources, e.g., in cases of realia collections, information management and service oriented institutions and research data collection, e.g., The Digital Archaeological Record [56] of Digital Antiquity [57].

For example, in focus cases of archaeology, prehistory, and history context and georeferencing can further be supported by facet creation into more dimensions and also allows the application of a consistent conceptual base for description and fuzziness, beyond common auxiliaries and georeferencing. Spatial references beyond common auxiliaries of place [23] (Table IV), especially 'Places of the ancient and mediaeval world' (UDC:(3)) and 'Countries and places of the modern world' (UDC:(4/9)) can be created from the following tables of UDC code references.

Tables X and XI provide example excerpts of relevant main conceptual knowledge and details of UDC references used for conceptual mapping. For conceptual knowledge of place and spatial context the implementations requires to provide references to classification codes. The UDC provides references based on the common auxiliaries of place of the UDC [23] as excerpted here for facets of place and space, physiographic designation, and places from ancient to modern world.

TABLE X. CONCEPTUAL MAPPING REFERENCES WITH UDC CODES OF SPATIAL FEATURES / PLACE: AUXILIARIES OF PLACE (EXCERPT).

UDC Code	*Description*
UDC:(1/9)	Common auxiliaries of place.
UDC:(1)	Place and space in general. Localization. Orientation
UDC:(100)	Universal as to place. International. All countries in general
UDC:(1-0/-9)	Special auxiliary subdivision for boundaries and spatial forms of various kinds
UDC:(1-0)	Zones
UDC:(1-1)	Orientation. Points of the compass. Relative position
UDC:(1-2)	Lowest administrative units. Localities
UDC:(1-5)	Dependent or semi-dependent territories
UDC:(1-6)	States or groupings of states from various points of view
UDC:(1-7)	Places and areas according to privacy, publicness and other special features
UDC:(1-8)	Location. Source. Transit. Destination
UDC:(1-9)	Regionalization according to specialized points of view

TABLE XI. CONCEPTUAL MAPPING REFERENCES WITH UDC CODES OF SPATIAL FEATURES / PLACE: PHYSIOGRAPHIC DESIGNATION (EXCERPT).

UDC Code	*Description*
UDC:(2)	Physiographic designation
UDC:(20)	Ecosphere
UDC:(21)	Surface of the Earth in general. Land areas in particular. Natural zones and regions
UDC:(23)	Above sea level. Surface relief. Above ground generally. Mountains
UDC:(24)	Below sea level. Underground. Subterranean
UDC:(25)	Natural flat ground (at, above or below sea level). The ground in its natural condition, cultivated or inhabited
UDC:(26)	Oceans, seas and interconnections
UDC:(28)	Inland waters
UDC:(29)	The world according to physiographic features
UDC:(3/9)	Individual places of the ancient and modern world

The references, e.g., classification, facets, concordances, and textual description, are usable in all the procedures and steps and allow to consider and implement arbitrary flexibility of fuzziness. Entry points to relevant and associated knowledge may be in any disciplinary context due to the consistent framework of the UDC and the multi-disciplinary and multi-lingual Knowledge Resources.

In our practical research projects and implementations, deploying a modular integration of consequent knowledge resources' components and their development with means of conceptual knowledge pattern matching has proven to enable valuable solutions for challenging and complex cases of in many disciplines.

X. CONCLUSION

This research achieved to implement the goal of integrating knowledge resources' development and universal conceptual knowledge pattern matching for practical context creation and georeferencing of archaeological, prehistorical, and geoscientific object entities. Respective module implementations, components developments, and results of automatically generated and consistent knowledge context and georeferencing were shown.

The information science integration case study demonstrated the practical components and their coherence for creating context and generating context results, e.g., visualisation, cartographic and topographic context including advanced solutions of object entity selection for georeferencing. It must be emphasised that this research targets an information science case study for multi-disciplinary resources and context, and cannot be limited to plain principles of case studies of technical or engineering aspects.

The implementations successfully employ the method of knowledge pattern matching based on the CKPM methodology. Further, the knowledge based mining implementation successfully employs the UDC references in order to provide the required conceptual framework. The UDC references proved to provide an excellent core component, for universal, multi-disciplinary, and multi-lingual knowledge.

In this new context, UDC showed to have a perfect organisational structure of conceptual knowledge for practical, systematical use as well as for an efficient and flexible processing support, following respective knowledge forks for references while creating and keeping developing resources and conceptional knowledge consistent supported by its editions.

The rewarding benefits of the methodology allow disciplines, e.g., archaeologists, to work with complex and multi-disciplinary and multi-lingual knowledge references in a consistent way, which can by far no longer be dealt with manually or even with anachronistic practices. The knowledge-based methodology enables addressing arbitrary knowledge and facets and to create new and individual methods for dealing with complements, based on common knowledge resources.

The general framework includes targetting previously unidentified references to knowledge resulting from the referenced complements. The methodology provides a wide range of facilities for individual methods to deal with application scenarios, multi-dimensional object references, fuzziness and arbitrary fine granularity of knowledge and references and even references from external notations.

However, each of the deployed implementation components, e.g., classification, phonetic algorithms, and topographic context data, and GMT mapping, may require intensive study and even years of experience. The integration may even require additional levels of understanding and experience. Therefore, to significant extent, the will and seriousness to invest in and work with information science methods is an unavoidable precondition for any discipline to create successful solutions for millions of references and objects based on the presented methods.

In daily practice, the integration based on the new method provides excellent and sustainable conceptual documentation and enables to create associations and links between knowledge object entities, which cannot result with comparable consistency otherwise. Further, configuring knowledge ranges can be achieved in many ways, e.g., by limiting resources, configuring the pattern depths and widths, ranking and selection.

Future research on theory and practice will continue developing practical conceptual knowledge frameworks, especially for prehistory and archaeology and natural sciences context, suitable knowledge resources, knowledge patterns, and further means of integration modules and components.

ACKNOWLEDGEMENTS

We are grateful to the "Knowledge in Motion" (KiM) long-term project, Unabhängiges Deutsches Institut für Multidisziplinäre Forschung (DIMF), for partially funding this research, implementation, case studies, and publication under grants D2018F6P04938 and D2018F3P04932 and to its senior scientific members and members of the permanent commission of the science council, especially to Dr. Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek (GWLB) Hannover, to Dipl.-Biol. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, to Dipl.-Ing. Martin Hofmeister, Hannover, and to Olaf Lau, Hannover, Germany, for fruitful discussion, inspiration, practical multi-disciplinary case studies, and the analysis of advanced concepts. We are grateful to Dipl.-Ing. Hans-Günther Müller, HPE, Germany, for his excellent contributions and assistance providing practical private cloud and storage solutions. We are grateful to the members of the Eastern Mediterranean research and studies campaign 2018–2020, DIMF, and all national and international partners in the Geo Exploration and Information cooperations for their constructive and trans-disciplinary support. We are grateful to The International ARS Science and History Network and The Science and High Performance Supercomputing Centre (SHPSC) for long-term support. / DIMF-PIID-DF98_007.

REFERENCES

[1] C.-P. Rückemann, "From Knowledge and Meaning Towards Knowledge Pattern Matching: Processing and Developing Knowledge Objects Targeting Geoscientific Context and Georeferencing," in Proceedings of The Twelfth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2020), November 21 – 25, 2020 - Valencia, Spain. XPS Press, Wilmington, Delaware, USA, 2020, pages 36–41, ISSN: 2308-393X, ISBN-13: 978-1-61208-762-7, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2020_3_10_30035 [accessed: 2020-08-09].

[2] M. Foucault, The Archaeology of Knowledge. Routledge Classics, 2002, ISBN: 978-0-415-28752-4, Translated by A. M. Sheridan Smith.

[3] Aristotle, On Interpretation, 2008, Peri Hermeneias, Organon, (Written 350 B.C.E.), Translated by E. M. Edghill Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Aristotle/interpretation.html [accessed: 2020-07-12].

[4] C. K. Ogden and I. A. Richards, The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, 1923.

[5] F. de Saussure, Cours de linguistique générale, 1916, (title in English: Course in General Linguistics), Charles Bally and Albert Sechehaye (eds.).

[6] L. W. Anderson and D. R. Krathwohl, Eds., A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Allyn & Bacon, Boston, MA (Pearson Education Group), USA, 2001, ISBN: 978-0801319037.

[7] Aristotle, The Ethics of Aristotle, 2005, Project Gutenberg, eBook, EBook-No.: 8438, Rel. Date: Jul., 2005, Digit. Vers. of the Orig. Publ., Produced by Ted Garvin, David Widger, and the DP Team, Edition 10, URL: http://www.gutenberg.org/ebooks/8438 [accessed: 2020-01-12].

[8] C.-P. Rückemann, F. Hülsmann, B. Gersbeck-Schierholz, P. Skurowski, and M. Staniszewski, Knowledge and Computing. Post-Summit Results, Delegates' Summit: Best Practice and Definitions of Knowledge and Computing, Sept. 23, 2015, The Fifth Symp. on Adv. Comp. and Inf. in Natural and Applied Sciences (SACINAS), The 13th Int. Conf. of Num. Analysis and Appl. Math. (ICNAAM), Sept. 23–29, 2015, Rhodes, Greece, 2015, DOI: 10.15488/3409.

[9] Plato, Phaedo, 2008, (Written 360 B.C.E.), Translated by Benjamin Jowett, Provided by The Internet Classics Archive, URL: http://classics.mit.edu/Plato/phaedo.html [accessed: 2020-01-12].

[10] C.-P. Rückemann, "Methodology of Knowledge Mapping for Arbitrary Objects and Entities: Knowledge Mining and Spatial Representations – Objects in Multi-dimensional Context," in Proceedings of The Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2018), March 25–29, 2018, Rome, Italy. XPS Press, Wilmington, Delaware, USA, 2018, pp. 40–45, ISSN: 2308-393X, ISBN: 978-1-61208-617-0, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2018_3_20_30078 [accessed: 2020-01-12].

[11] C.-P. Rückemann, "Superordinate Knowledge Based Comprehensive Subset of Conceptual Knowledge for Practical Mathematical-Computational Scenarios," in The Ninth Symposium on Advanced Computation and Information in Natural and Applied Sciences (SACINAS), Proceedings of The 17th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 23–28, 2019, Rhodes, Greece, Proceedings of the American Institute of Physics (AIP), AIP Conference Proceedings. AIP Press, Melville, New York, USA, 2020, ISSN: 0094-243X, (to appear).

[12] C.-P. Rückemann, Sustainable Knowledge and Resources Management for Environmental Information and Computation. Business Expert Press, Manhattan, New York, USA, Mar. 2018, Ch. 3, pp. 45–88, in: Huong Ha (ed.), Climate Change Management: Special Topics in the Context of Asia, ISBN: 978-1-94784-327-1, in: Robert Sroufe (ed.), Business Expert Press Environmental and Social Sustainability for Business Advantage Collection, ISSN: 2327-333X (collection, print).

[13] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: http://www.udcc.org/udcsummary/php/index.php [accessed: 2020-01-12].

[14] "Universal Decimal Classification Consortium (UDCC)," 2020, URL: http://www.udcc.org [accessed: 2020-07-05].

[15] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: http://creativecommons.org/licenses/by-sa/3.0/ [accessed: 2020-01-12], (first release 2009, subsequent update 2012).

[16] "Perl Compatible Regular Expressions (PCRE)," 2020, URL: https://www.pcre.org/ [accessed: 2020-08-09].

[17] "The Perl Programming Language," 2020, URL: https://www.perl.org/ [accessed: 2020-08-09].

[18] "UDC Summary Linked Data, Main Tables," 2018, URL: https://udcdata.info/078887 [accessed: 2020-01-12].

[19] "UDC, Common Auxiliary Signs," 2019, URL: https://udcdata.info/078885 [accessed: 2020-01-12].

[20] "UDC 9: Geography. Biography. History," 2019, URL: http://udcdata.info/068076 [accessed: 2020-01-12].

[21] "UDC 5: Mathematics. Natural sciences," 2019, URL: http://udcdata.info/025403 [accessed: 2020-01-12].

[22] "UDC ". . .": Common auxiliaries of time," 2019, URL: http://udcdata.info/011472 [accessed: 2020-01-12].

[23] "UDC (1/9): Common auxiliaries of place," 2019, URL: http://udcdata.info/001951 [accessed: 2020-01-12].

[24] C.-P. Rückemann, R. Pavani, B. Gersbeck-Schierholz, A. Tsitsipas, L. Schubert, F. Hülsmann, O. Lau, and M. Hofmeister, Best Practice and Definitions of Formalisation and Formalism. Post-Summit Results, Delegates' Summit: The Ninth Symposium on Advanced Computation and Information in Natural and Applied Sciences (SACINAS), The 17th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 23–28, 2019, Rhodes, Greece, 2019, DOI: 10.15488/5241.

[25] A. Bäck, Aristotle's Theory of Abstraction. Springer: Cham, Heidelberg, New York, Dordrecht, London, 2014, ISBN: 978-3-319-04758-4, ISSN: 1879-8578, The New Synthese Historical Library, (Book Series), Texts and Studies in the History of Philosophy, Volume 73.

[26] L. Učník, I. Chvatík, and A. Williams, The Phenomenological Critique of Mathematisation and the Question of Responsibility: Formalisation and the Life-World. Springer, 2015, ISBN: 978-3-319-09827-2, (Collection), Contributions to Phenomenology, Volume 76.

[27] W. H. S. Jones, Pausanias Description of Greece. London: William Heinemann, New York: G. P. Putnam's Sons, MCMXVIII, 1918, vol. I and II.

[28] A. Michaelis, Der Parthenon. Leipzig, Druck und Verlag von Breitkopf und Härtel, 1871, (title in English: The Parthenon).

[29] M. A. Tiverios, "Observations on the East Metopes of the Parthenon," American Journal of Archaeology, vol. 86, no. 2, 1982, pp. 227–229.

[30] K. A. Schwab, Celebrations of Victory: The Metopes of the Parthenon. Cambridge, Cambridge University Press, 2005, pp. 159–198, in: Jenifer Neils (ed.), The Parthenon: From Antiquity to the Present.

[31] C.-P. Rückemann, "Archaeological and Geoscientific Objects used with Integrated Systems and Scientific Supercomputing Resources," International Journal on Advances in Systems and Measurements, vol. 6, no. 1&2, 2013, pp. 200–213, ISSN: 1942-261x, URL: http://www.thinkmind.org/download.php?articleid=sysmea_v6_n12_2013_15 [accessed: 2020-08-09].

[32] R. C. Russel and M. K. Odell, "U.S. patent 1261167," 1918, (Soundex algorithm), patent issued 1918-04-02.

[33] D. E. Knuth, The Art of Computer Programming: Sorting and Searching. Addison-Wesley, 1973, vol. 3, ISBN: 978-0-201-03803-3, OCLC: 39472999.

[34] National Archives and Records Administration, "The Soundex Indexing System," 2007, 2007-05-30, URL: http://www.archives.gov/research/census/soundex.html [accessed: 2013-05-26].

[35] M. Stok, "Perl, Soundex.pm, Soundex Perl Port," 1994, (code after Donald E. Knuth).

[36] "LX SNDX, a Soundex Module Concept for Knowledge Resources," LX-Project Consortium Technical Report, 2013, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/#LX [accessed: 2020-07-12].

[37] E. Rempel, "tcllib, soundex.tcl, Soundex Tcl Port," 1998, (code after Donald E. Knuth).

[38] A. Kupries, "tcllib, soundex.tcl, Soundex Tcl Port Documentation," 2003, (code after Donald E. Knuth).

[39] "Global Land One-kilometer Base Elevation Project (GLOBE)," National Geophysical Data Center (NGDC), National Centers for Environmental Information (NCEI), National Oceanic and Atmospheric Administration (NOAA), NOAA Satellite and Information Service (NESDIS), U.S. Department of Commerce (DOC), URL: http://www.ngdc.noaa.gov/mgg/topo/globe.html [accessed: 2020-07-12].

[40] "2-Minute Gridded Global Relief Data (ETOPO2v2)," 2006, June, 2006, World Data Service for Geophysics, Boulder, USA, National Geophysical Data Center, National Centers for Environmental Information (NCEI), National Oceanic and Atmospheric Administration (NOAA), URL: http://www.ngdc.noaa.gov/mgg/fliers/06mgg01.html [accessed: 2020-07-12].

[41] C. Amante and B. W. Eakins, "ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis," 2009, NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center, NOAA. DOI: 10.7289/V5C8276M.

[42] "CGIAR Consortium for Spatial Information (CGIAR-CSI)," 2020, URL: http://www.cgiar-csi.org [accessed: 2020-07-12].

[43] "Consultative Group on International Agricultural Research (CGIAR)," 2020, URL: http://www.cgiar.org [accessed: 2020-07-12].

[44] "World Geodetic System (WGS)," 2012, National Geospatial-Intelligence Agency (NGA), URL: http://earth-info.nga.mil/GandG/wgs84/index.html [accessed: 2020-07-12].

[45] "Spatial reference for EPSG:4326," European Petroleum Survey Group Geodesy (EPSG), URL: https://epsg.io/4326 [accessed: 2020-07-12].

[46] "GMT - Generic Mapping Tools," 2020, URL: http://gmt.soest.hawaii.edu/ [accessed: 2020-07-12].

[47] R. Gleser, "Beitrag zur Klassifikation und Datierung der palmettenförmigen Gürtelschließen der späten Latènezeit," Archaeological Correspondence (in German: Archäologisches Korrespondenzblatt), Bd. 34, vol. 34, 2004, pp. 229–242, (title in English: Contribution for Classification and Dating of Palmette-shaped Belt Buckles in the Late Latène).

[48] R. Gleser, O. Al-Saadi, V. Schmidt, and T. Fritsch, "Geophysical Prospection and Archaeological Excavation at a Celtic Burial Site near Nonnweiler, Saarland," 2015, Poster presentation in the session 'Archeogeophysics' (in German: 'Archäogeophysik') (AR) at the 75th annual meeting of the German Geophysical Society (in German: Jahrestagung der Deutschen Geophysikalischen Gesellschaft) (DGG) in Hannover, 2015.

[49] R. Gleser, "Ein Technologiesprung - Frühes Metall. Wissen, Funktion und Symbol," in: T. Otten, J. Kunow, M. M. Rind, and M. Trier (Eds.), Revolution jungSteinzeit. Katalog zur Archäologischen Landesausstellung Nordrhein-Westfalen. Schriften zur Bodendenkmalpflege in Nordrhein-Westfalen, Bd. 11,1 (Darmstadt 2015), vol. 11,1, 2015, pp. 251–259, (title in English: A Technological Leap – Early Metal. Knowledge, Function, and Symbol).

[50] R. Gleser, "Neue Überlegungen zur Chronologie der postbandkeramischen Kulturphänomene in Mitteleuropa," in: Centenary of Jaroslav Palliardi's Neolithic and Aeneolithic Relative Chronology, International Conference Moravské Budejovice 2014, 2016, pp. 107–116, Moravské Budejovice 2016, (title in English: New Considerations on the Chronology of Post Band Ceramics Cultural Phenomena in Central Europe).

[51] R. Gleser, Zu den erkenntnistheoretischen Grundlagen der Prähistorischen Archäologie. Leiden, 2020, 2020, (title in English: On the Epistemological Fundaments of Prehistorical Archaeology), in: M. Renger, S.-M. Rothermund, S. Schreiber, and A. Veling (Eds.), Theorie, Archäologie, Reflexion. Kontroversen und Ansätze im deutschsprachigen Diskurs, (in print).

[52] R. Gleser, Rekonstruktion der Vergangenheit: Zur methodischen Eigenart prähistorischen Erkennens. Paderborn, 2018, pp. 199–237, (title in English: Reconstruction of the Past: On the Methodical Peculiarity of Prehistorical Cognition), in: A.-S. Naujoks, J. Stelling, and O. R. Scholz (eds.), Von der Quelle zur Theorie. Vom Verhältnis zwischen Objektivität und Subjektivität in den historischen Wissenschaften, ISBN: 978-3-95743-136-3.

[53] R. Filloramo, V. Becker, and A. Curci, "Apulian prehistoric community connections: Preliminary results of GIS analysis and field activity," Journal of Archaeological Science: Reports, vol. 32, 2020, pp. 1–11, 102380, URL: https://doi.org/10.1016/j.jasrep.2020.102380 [accessed: 2020-11-08].

[54] V. Becker, Gemeinsamkeiten und Unterschiede bildlicher Darstellungen zwischen Jungpaläolithikum und Neolithikum. Münster, 2018, 2018, pp. 3–13, (title in English: Commonalities and Differences of Figurative Depictions Among Late Palaeolithic and Neolithic), in: T. Schreiber (Ed.), Zugänge zur Kontextualisierung bildlicher Darstellungen in den Altertumswissenschaften. Workshop des Netzwerks Archäologie Diagonal. 28.–29. Juni 2013 an der WWU Münster, URL: https://repositorium.uni-muenster.de/transfer/miami/2cbdabb9-e908-4b8f-9f4f-1745ea06e8cd [accessed: 2020-11-08].

[55] V. Becker, "Keramik der Lausitzer Kultur aus der Lehrsammlung der Abteilung für Ur- und Frühgeschichtliche Archäologie der Universität

Münster," 2012, (title in English: Ceramics of the Lausitz Culture from the Collection of the Division of Prehistoric Archaeology, University of Münster), URL: https://www.uni-muenster.de/imperia/md/content/geschichte/ufg/pdf/lausitzer_kultur.pdf [accessed: 2020-11-08].

[56] "The Digital Archaeological Record (tDAR)," 2020, URL: http://www.tdar.org [accessed: 2020-08-05].

[57] "Center for Digital Antiquity," 2020, Arizona State Univ., URL: http://www.digitalantiquity.org/ [accessed: 2020-08-05].

Managing Patient Records in Complex Systems in Time of Epidemics

The Predominence of Human Aspects for Quality of Processes

Eloria Vigouroux-Zugasti, Christian Bourret
Laboratory of DICEN IDF
University of East-Paris Marne-la-Vallée
Val d'Europe / Serris, France
Email: eloria.vigouroux-zugasti@univ-eiffel.fr,
christian.bourret@unive-eiffel.fr

Sylvie Parrini-Alemanno
CNAM
Paris, France
Email: sylvie.parrini-alemanno@lecnam.net

Abstract — **With the recent promulgation of the General Data Protection Regulations, data management is becoming a crucial strategic issue in organizations. The quality of data dissemination is of utmost importance in the healthcare environment. Indeed, medical confidentiality is closely linked to the dissemination of personal information inherent in the patient's record. Yet, how can a complex system, composed of multiple multidisciplinary actors (medical, paramedical, administrative, etc.), deal with the potential disclosure of personal data? What steps can be taken to manage this risk? With the current health crisis of the COVID-19, those questions become essential. Medical confidentiality and crisis management has to deal with massive ethical issues related to the use of personal data. The interoperability of professionals and the quality of care are more crucial than ever, considering the need to limit the spread of the virus. To answer these questions, a case study was conducted in the Multidisciplinary Care House of Mimizan (France, Landes, New Aquitaine Region). The goal was to investigate the importance of human and data management for traceability of care and for crisis management. This medical organization is composed of medical and paramedical professionals, but also a relatively large administrative team for such an institution. The results highlight that this organization manages to set up, at the initiative of the professionals, both flexible and structured processes, allowing optimal follow-up of patients, while guaranteeing respect for their personal information.**

Keywords — Health; Data; Patient Records; Quality; Crisis; Human Relationship; COVID-19; Territory; Organization.

I. INTRODUCTION

The management of personal data, with the recent promulgation of the General Data Protection Regulations (GDPR - May 2018), becomes a crucial and strategic issue for organizations. In the healthcare sector, respect for medical confidentiality is closely linked to the dissemination of patients' personal records. Data has to be managed with great care in order to limit the risk of unfortunate disclosure or data loss. Data management in health organizations depends on both technical and human factors, which are closely related to each other. Technical aspects (storage, access, sorting, etc.) would only be efficient if they are linked to a human data management policy (uses guidelines, access ethics, conditions of diffusion inside and outside the organization, etc.).

However, the current French healthcare organization tends to evolve to a collaborative working method, as some researchers have been able to show in their studies [1]. For the past ten years, French health authorities have been witnessing the emergence of some new organizations of health professionals and the exponential growth of groups of (para)medical professionals, attesting the growing complexity of health organizations [2]. This is particularly the case for Multidisciplinary/Multiprofessional Health Houses (MCHs), with 910 establishments active in France in March 2017, compared to only 240 in 2013 [3]. These establishments are complex because of the number of interacting actors they involve. They also tend to rely on the notion of sharing and circulating data, especially health information about patients. In doing so, they tend to improve patients' health care and health monitoring.

The current health context tends to intensify the difficulties in the management of health in France - and *a priori* in the whole world also. This is why reflections and practices related to the management of personal records, especially from a human point of view, must be intensified. Indeed, health professionals are under intense pressure. Lack of trust in the organization, organizational misfunction, team tensions, exhaustion, stress and fear of being contaminated represent high risk factors that could lead to (un)voluntary misuse of personal records. For health organizations, it is critical to find solutions to preserve their quality. It is also critical to remain scrupulous with the protection of data crisis in order to guarantee the quality of care, while protecting patients' records and especially those who are contaminated.

How could both technical and human processes guarantee medical confidentiality and manage the inherent risks linked to data disclosure, while ensuring the interoperability of professionals and the quality of care, in time of crisis? It seems that the challenge is mainly about the ability of the French health system to increase its level of performance [4]. The challenge also rests on the ability of collaborative organizations to hold their staff together, despite the pressure they have to work with.

To answer this question, a first case study was conducted in the Multidisciplinary Care House of Mimizan (MCHM). To do so, an exploratory qualitative approach was conducted via focus groups with the whole team of the institution and via an interview with its two managers. The goal is to establish an inventory of practices, particularly in terms of data quality management in complex health institutions. The analytical approach presented here is based on the structural level of MCH actions, in which resources (human and material) are mobilized to ensure a good coordination for patients' care and confidential data policy [5]. MCHs are a recent and emergent phenomenon, with heterogeneous ways of working. The goal of this article is to explore one of the biggest and formalized MCH, from an exploratory point of view, in order to understand its organization and its capacity to be considered as a model for other MCH. A second phase of the survey was conducted following the same methodology, when the French government declared the state of emergency to protect the population from the COVID-19 epidemic, on March 12, 2020. The goal is to compare the ordinary processes to the crisis ones and to deepen the human aspects of data management.

Section II is dedicated to the overall contextualization of the study. Section III presents the institution selected for the study, the MCHM to show how it is representative of the new healthcare institutions needed in fragile territory. In Section IV, the article analyzes the quality processes inherent in the management of this institution's data. The Section V presents the specifications of the MCHM. Section VI provides details on the general lack of use of the national digital health record. Sections VII to IX are dedicated to the second phase of the survey, related to the adaptation of the MCHM to the COVID-19 health crisis. Section VII concerns the hypertelic aspects [6] of organizational processes in the field of public health – that is, organizational processes hyper-adaptive to their context but wich may become impossible to adapt if the context changes. In Section VIII, the evolution of organizational management is developed. Section IX presents the model of collaboration and interdisciplinary cooperation as a necessary step to succeed in time of crisis. We bring a conclusion of our analysis in Section X.

II. Study Background

In this section, the study background is explained. The specific context of the current French public healthcare context and policy are also detailed for a better understanding of the study.

A. *Challenging health context*

In 2014, the population density of Landes department fluctuated between 2 and 45 inhabitants/km², with an aging index among the highest in the region [7]. In 2015, 31.5% of Landes' population was over 60 years of age, a great increase compared to 2011 [8]. According to the Regional Health Authority – RHA (in French, *Agence Régionale de Santé*), in 2016, the rural population represented more than 50% of the whole Landes population. In addition, this region also has a medical demography and a density of specialists lower than regional and national averages, as well as a small number of healthcare establishments. It also has few alternatives to the nursing homes for old people [7]. Therefore, old people's loss of autonomy is more difficult to manage. According to RHA, 1/3 of liberal general practitioners were over 60 years old in 2017 [9]. This is a very problematic issue: combined with the difficulties of attractiveness of the territory, it becomes more and more difficult to maintain the number of practitioners in this area. The RHA demographic patterns of Landes health care show a highly unfavorable public health context:

- White areas (towns located more than fifty kilometers away from a hospital emergency department), combined with an insufficient number of expert services (radiology, rheumatology, gynecology, allergology, pneumology, dermatology, etc.)
- Medical desertification: unattractive territory for young (para)medical professionals (region's remoteness from large cities, low internet coverage, few cultural offers, etc.). Doctors are struggling to find successors, despite administrative provisions and facilitations offered by health authorities.
- Fragile areas: unequal distribution of health professionals in areas with an imbalance between the number of potential patients and the number of doctors, as well as areas where the advanced age of patients (or the doctor) would require urgent decisions.

Facing these difficulties, some political representatives try to shed some light on the issue of medical deserts: in October 2018, the mayor of a small town called Ychoux proposed to prohibit, by municipal decree, his fellow citizens from falling ill, due to the lack of medical care in its surroundings. The study was conducted in this area for its relevance concerning the challenges France will have to face in the coming years. The main problem in France is getting young medical staff to settle in countryside areas, where living standards are less attractive than urban contexts.

B. *Management of de-materialized health data in a complex system*

Personal records, such as medical data, defined by the French Data Protection Act, called *Loi informatique et liberté*, (cf. paragraph 2, article 2 of the Act) as an information relating to a person who is physically identified or who can be identified, directly or indirectly, by reference to an identification number or to one or more elements specific to him/her [10]. In addition, this data must be processed in order to make sense. This process is defined by the same law (cf. paragraph 3, article 2 of the Act) as a transaction or set of transactions, whatever the process used [10]: collection, recording, organization, storage, modification, consultation, communication by transmission, etc. Data are also part of an exchange, characterized by the provision to several professionals, such as health staff, who are entitled to know everything about these data. Their goals are to insure the coordination and continuity of patients' medical care [11].

However, the exploitation of personal records remains a sensitive subject. It directly affects the privacy of each individual [12], especially when it comes to medical data. In complex systems, such as MCH, there are several issues related to the management and to the protection of health data. From an organizational point of view, it is essential to set up procedures to secure access to data. Those procedures tend to limit the structural disorders that can affect the confidential standards of data, by establishing, for example, quality indicators. Measurement and management tools in health establishments are essential [4]. In addition, complex systems have a large number of stakeholders, but they do not have the same level of data access authorization. It increases the risks of fraudulent or accidental access to information.

With the multiplication of MCHs, the French health sector must now face a multitude of risks related to data, which require close scrutiny of each elementary activity [2]. These establishments are the result of a clustering of health professionals who, until now, had been working alone. However, using common resources and administrative staff lead professionals to rethink their working methods, while insisting on control and rigour. The various stakeholders in the project have to develop fundamental procedures for collaborative work. This aims at reducing and optimizing work processes [2], while considering the topics of control and quality as the heart of these processes. Professionals, in this new context, must demonstrate that their services are delivered in a secure environment. This environment helps controlling the risks and meet the expectations/requirements of patients [4].

C. *Multidisciplinary health centre: although need for a restructured health policy – from a political point of view*

MCHs are a model that catalyzes needs, from health professional, public decision makers and patient care points of view [5]. The creation of this type of institution represents the convergence of three complementary processes, identified by Autès and Dufay [13]:

- Movement initiated by healthcare professionals to gather their activities within MCHs and health centers.
- Reflection of local officials, concerned by the management of health in their districts, involved in logics of prevention, of permanence of care, of first aid and the continuum between outpatient services and hospitals. They also care about offering external and specialized consultations to the people living in areas where there is a shortage of health practitioners;
- Necessary reorganization of the supply of care due, first, to the constraints of modern medicine and pathologies and, second, to the effects induced by the anticipated decline in medical demography.

From a territorial point of view, the MCHM aims at meeting the four standards of public health action: 1) maintaining a local offer, 2) guaranteeing equal access to health for all, 3) ensuring continuity of care between the primary care offer and graduated hospital care 4) and, finally, strengthening health prevention policies. For local officials, the issue is to strengthen weakened health districts and care offer [13], by proposing long-term ways-out to solve the current problems. In addition, since MCHs are subjected to accreditation rules, by responding to quality indicators established by public health authorities [13], they contribute to an increased performance of districts' medical management.

III. Methodological Approach

In this section, the sampling and methodological procedures are presented.

A. *Why Study the MCHM?*

By definition, MCHs depend on specific contexts. It is important to identify the territory's needs, to take into account the needs of its population and its state of supply. MCHM is considered as representative of this movement of territorial restructuring in the field of healthcare both in its conditions of implementation and in its daily functioning, structured around the interrelationships between territorial stakeholders [5]. Indeed, despite its recent implementation, it manages to meet the whole public health objectives, both mandatory and optional, imposed by the "RHA Inter-professional Agreement" contract, particularly in terms of shared information systems, which are at the heart of the challenges related to data quality. The dynamic of the creation of the MCHM was, in the first place, launched by healthcare practitioners themselves, in reaction to the progressive desertification of their territory and the challenges it involves [13]. The project of creating the MCHM began in 2004. The district's doctors wanted to cluster their activities in a single establishment, to pool their administrative tasks and to offer a better access to care for their patients. This approach is in line with the observations of some researchers, who state that medical desertification in rural areas has been the main motivation for the mobilization of health professionals [13]. One of the main problems lays in Landes' unattractiveness for doctors. The mere proximity of the beach and the "sweetness of life" are not enough to attract young professionals willing to settle down. It is necessary to provide health professionals with some attractive and secure professional conditions of practice.

However, the notion of attractiveness of the project is very important here [5]. Offering a young professional an isolated practice in a small town does not have the same appeal as a long-term position in a MCH, in which he or she could be supervised, advised and supported by administrative services, surrounded by colleagues and supported by financial and material resources. Collegiality and plurality of perspectives make the medical practice both more reassuring and richer [13], especially at the beginning of a career. This is the appeal proposed, in general, by MCHs and, in particular, by the case studied in this article, which is one of the biggest and dynamic MCHs in France. Its professionals have been recently asked to present their institution in the next National French Congress of MCH. This type of organizational dynamism is a movement widely desired and claimed by the younger generations of medical and paramedical professionals [14].

Moreover, Landes is the largest region of France, with a mainly rural territory and offers most of the current and forthcoming public health services presented in the contextualization part of this article. Having such an innovative MCH in this kind of area is an example of how to deal with public health issues in other regions in France, especially concerning the rural ones.

B. Methodological Approach and Sampling

The survey was conducted from the 8th of December 2018 to the 10th of March 2019. This period provides two main contexts: seasonal epidemics (the flue in January and spring allergies in March) and quiet period in December and February. It seemed relevant to investigate the MCHM in both these contexts in order to deepen all the aspects of its organizational data management. At the beginning of the survey, the goal was to identify the organizational model and rules implemented by the MCHM's team, specially concerning working processes and data management. To do so, a qualitative approach has been chosen This method has been selected for its ability to investigate the practices and interpret the results. It considers that the "confrontation with the corpus is a necessary condition for the perception of social practices" [14]. The goal was also to confront the different points of view concerning the organizational processes. The potential divergences and discordances regarding the positions can be highlighted. Does a secretary think the same thing of the establishment than a doctor or paramedical worker? Concerning data privacy management, the heads of the MCHM in charge have been interviewed. They had to explain their choices in terms of data management policy, of coordination put in place and of emergency plans in case of unfortunate disclosure.

Regarding the questions, all the members of the MHCM staff we asked about two common topics. The first was about the daily-work and its organization, both concerning the inner-group and the relationships with the other members of the MCHM (for example: secretary-secretary, secretary-medical, etc.). The goal was to highlight relational and organizational dysfunctions. Secondly, all the staff members were asked about their own professional uses of patients' health records, in terms of access of use and of transmission. The purpose was to identify good and problematic uses.

The second questions asked for some more specific topics. The goal was to have a better understanding of each specific staff members (medical doctor, paramedical, administrative, etc.), to point out the benefits and the limits of their new work, management and organizational processes since they entered the MCHM. Four focus groups took place in December 2018 and February 2019, with the four specific staff members. Then, interviews were conducted in January 2018 and March 2019, with the head of the administrative staff and the heads of the MCHM. They all were realized in the MCHM, in the meeting room. The goal was to make the people feel comfortable and to prevent conversations from being heard by the patients or the other staffs' members. This approach seemed relevant, as it helped people to speak freely.

A second phase of the survey was conducted during the COVID-19 crisis, from the 12th of March 2020 to the 11th of May 2020, in the same organization. This period corresponds to the riskiest period of the pandemic, from a public health point of view, when France reached its highest rates of virus spread. It also corresponds to the most difficult time for public health organizations, which have had to adapt urgently their processes to face this situation they never knew before. For the sake of homogeneity, it was decided to use the criteria and distribution initially used for the first phase of the survey in order to compare the results. It was also decided to conduct the interviews by teleconference. The goal was to limit the risk of exposure to the virus and to make the survey as smooth as possible for the MCHM staff.

C. Health Records Management in the MCHM

TABLE I. DISTRIBUTION OF THE INTERVIEWED SAMPLE

Criteria	Distribution	Number	%
Gender	Male	14	51.9
	Female	13	48.1
Age	25-35	6	22.2
	35-45	7	25.9
	45-55	8	29.7
	55-65	6	22.2
Professional activity	Medical doctors	11	40.7
	Paramedical staff	8	29.7
	Administrative staff	5	18.5
	Executive of administrative staff	1	3.7
	Executive of the MCHM (also doctors)	2	7.4

As Table I shows, the MCHM brings together three main crews: doctors (8 + 2 regular substitutes + 1 trainee), paramedics (3 nurses and their collaborators, 2 physiotherapists, 2 podiatrists, 1 psycho-motor therapists, 1 dietician) and administrative staff (5 secretaries, only working for doctors, managed by an executive). Including trainee doctors and nursing staff, the sample is composed of thirty individuals involved in the daily operations of the MCHM. They are all subjected to the institution's collective agreement, in which respect for medical confidentiality is clearly enshrined. Data is managed and stored on a specific medical database software, Weda [15], which is also used by the surrounding external collaborators (pharmacies, specialists, hospitals, etc.). The choice to use the same software aims at facilitating the medical and administrative aspects concerning the caring continuum. Each member of the MCHM has secure access provided by the software. These codes are not stored on the institution's digital devices in order to limit the risk in case of theft or hacking.

However, not all MCHM members have the same level of access to patients' records. Doctors, as well as their secretaries, have full access to all the information concerning: files, auxiliary session schedules, secure messaging, etc. This is justified by the need for doctor/doctor and doctor/secretary interoperability, for the smooth functioning of the MCHM and good patient care. At the request of a patient or a staff member, restrictions may be applied to limit secretaries' or doctors' access to some records. All patients were asked about this sharing consent.

Then come the paramedics, with a diffusion specific to each specialty. Their access is conditioned by the needs of their activity. A physiotherapist, for example, will have access to the patient's x-rays and related prescriptions; a nurse will have access to history, specialist contacts, blood test results or vaccines, depending on needs. These professionals do not have access to the content of visits, letters, prescriptions, unless some specific case discussed with the doctor. The accreditation of external professionals is aligned with the system applied within the institution, according to the needs of the patient, the activity or the specialty. Collaborative patient follow-up is governed by Multi-Professional Consultation Meetings (MPCMs), planned or impromptu, attended by all the professionals involved in the presented case. Each meeting is documented, stored via Weda and only accessible to the concerned professionals. Doctors and secretaries meet weekly to monitor performance and improve organizational quality processes. The doctors interviewed also associate these meetings with team management (trust, accountability, etc.), to ensure cohesion among all workers and to involve them in the administration of the MCHM.

This institution is a motivated and voluntary grouping of health professionals, in a singular and innovative organizational form. As a result, this configuration will have to be closely observed in the years to come in order to evaluate its lean management in data management and its mode of transactional relations, perceived as co-constructive interactions.

IV. Analysis of Data Management Processes

This section is dedicated to the analysis of MCHM data management concerning patients' records. Thanks to the study, the main protagonists of this type of management tasks have been identified, such as the governance of the establishment, from a managerial point of view.

A. *Data Ethics and Dissemination Quality*

An organization can be defined as a set of recurring transactional programs that constitute transactional flows. They are driven by a set of conventions and rules in a given context [16]. For its proper functioning, given the complexity of its transactions, it is essential to give access to the right information, at the right people, at the right time, to make a selective transmission of users, in order to fight against misinformation, over-abundance and deviant uses [12]. The principle of data management is based on the ability of actors to select information and analyze it, in such a way that it is only disseminated to its legitimate recipients. The interest of this approach is twofold. On the one hand, it allows a smooth organization of sharing actions, making the institutional processes efficient. On the other hand, it makes it possible to limit the risks linked to the poor dissemination of data, thus guaranteeing respect for confidential medical records.

The ethical processing of information seems to be the starting point of the MCHM's data management strategy. Béranger defines it as a mechanism for the interpretation of data, by a person or an organization, that will lead to give a specific meaning to data [12]. By giving attention to information, by analyzing it, the heads of MCHM tend to give meaning and value to data, as well as to determine the logistics of action to be applied: censorship, global dissemination, limited dissemination, etc. In the medical sector, it is fundamental to establish a reflection on personal health data through an ethical prism "in order to [remove] doubt and control uncertainties" [12] and to manage the risks inherent in the nature of patients' records. It leads to speak of the non-maleficence nature of the MCHM's information strategy: access to data is examined according to the profile and nature of the user [12]. Data sharing is conditioned by the profile of the information receiver, ranging from full sharing to very limited access, depending on activity and needs. This improves the security, confidentiality and protection of such data [12], as well as the performance of the information management system. By analyzing the data, determining the conditions for sharing and clearly identifying the receivers, the quality of access to patients' personal records is guaranteed.

B. *Informational Lean Management: no Unnecessary Information*

The data processing method leads us to analyze the notion of lean in quality management. Lean School is defined as "the search for process optimization by chasing down everything that is inappropriate or superfluous" guaranteeing "performance by eliminating waste" [2]. This method is usually applied to inventory management (0 stock), document management (0 paper), or logistics (0 unnecessary transport, 0 waiting, etc.). This can be relied to information management, in order to analyze the transaction rationalization activities [17].

Indeed, the info-ethical treatment as previously mentioned tends towards a very low entropy, i.e., a degree of almost nil disorder [12]. A system in which information is transmitted without analysis increases the level of confusion, as well as the slowness of decision-making and the risks of accidental dissemination of personal records. On the contrary, in a complex system such as the MCHM, the implementation of a hierarchy in information management (doctors analyze and choose the criteria before disseminating information) makes the actions of all team members easier and more fluid, by sending them only the data that will be useful to them in the exercise of their activity. This is a kind of lean management, applied to information management. This data dissemination method tends towards the goal of "0 useless information", in order to guarantee both respect for medical confidentiality (0 information poorly disseminated), the quality of patient care (0 information missing) and the fluidity of actions (0 dysfunction linked to poor information dissemination). This information management method seems to be perfectly adapted to the performance requirements of MCHM's missions, while benefiting the daily tasks (administrative and patient care).

V. Organizational processes: guaranteeing the quality of data management

In this section, the organizational processes observed during the survey will be discussed, which refers to data and patients' records management.

A. The Human Relations Theory as a Leading Light

In the MCHM, the mobilization of the whole staff tends to improve the processes' efficiency [2]. The MCHM's management method is based on the involvement of all teams in improving the life of the institution: meetings, taking into account opinions, professional development, empowerment, etc. This method seems to be similar to the collaborative processes set up within the MCHM, although the institution does not claim any particular managerial method: unexpected discussions, weekly team meetings, festive group cohesion events, etc. This team management aims at analyzing defects and dysfunctions, and then seeking solutions [2]. This tends to improve the overall functioning of the establishment, where, according to Zacklad, all persons involved in the transaction are in the position of (co-)director, (co-)beneficiary, (co-)recipient (principal) and (co-)recipient [16]. Emphasis is placed on freedom of speech, professional responsibilities, skills of each individual and, above all, the necessary trust between employees, which is considered essential by all MCHM staff.

As Doucet points out, it is essential for the direction of quality action to be collegial. This makes it possible [in particular] to respect responsibilities and involve departments: medical consultation, scheduling, data management, human resources, etc. [2]. As a large number of individuals have access to the institution's health data, the use of this collaborative and collegial approach is essential to the MCHM. The increase in performance can only be achieved through the collaboration with the departments involved in this approach. By soliciting and valuing all staff members, the MCHM ensures fine relationship management, but also encourages professionalism and accountability of each individual. They also do so by regularly reminding them of the need for secrecy and rigor (formally and informally), especially concerning the performance of their daily tasks relating to patient health records. Transactional relationships lead to overcome formal/informal oppositions by insisting on their complementarity [5].

B. Leadership: Team, Quality and Performance Management

This managerial approach is in line with the objective of promoting confidentiality and trust with producers and suppliers of information, thus contributing to the control of risks and deviations of data [12]. However, for MCHM staff management to be effective, it must also deal with some leadership issues. Leadership and management of the institution must be provided by a person of influence who, thanks to his or her managerial skills is able to guarantee, effective cooperation and coordination, based on mutual trust [5]. The notions of cooperation and trust seem to be interdependent key resources for the management of complex systems. Cooperation relies on a clear commitment of each member of the group and is strengthened by trust and by the working contract [17]. In a complex system, each member of the team contributes to the success of the institution goals, so it is important for all team members to be aware of the values their work involves, such as secrecy, efficiency, empathy, etc. [18] It is up to the leader to make the team understand these fundamental values defended by the institution, from which ethics rules of behavior flow. To do so, in the MCHM, many meetings are held, with the whole staff or with some subgroups (doctors-doctors, doctors-secretaries, etc.). Managing does not mean dominating. It is rather knowing how to talk to teams and how to get them to work towards a common goal [19]. Even if some members sometimes complain about the high number of meetings, they seem essential for the good management of the institution. Those meetings allow the team to have some feedback on the work and outline all technical or relational issues. It helps the manager to resolve the disagreements before they worsen and help the team to work with fluency [19].

The operational management of the MCHM is based on the involvement of the two doctors, whom will here be called P. and T. Their influence is based on their legitimacy within the team, gained through their seniority, their involvement in the project and their ability to organize the run of the institution. The team trusts in them. Trust is built over time and in the relationships. It is a capital that the two doctors accumulated through years [18]. They are well known by all the team members, sometimes for more than twenty years. They are also known for their emotional competencies [19], that combines feeling with objective cooperation skills. The long-term trust of the team gives P. et T. the ability to engage and influence each member of the group, which helps the team to solve complex issues [20] and aim for an outstanding performance of their work. P. and T. are complementary, both in terms of relational aspects and in the conduct of data establishment and management. However, for a good cohabitation, the roles must be clear and non-antagonistic [2]. The risk associated is the disappearance of authority representatives. The smooth running of the institution is based on the clear identification of authority figures, to which the staff can refer. The figure of authority also allows to the control of practices, beyond the "self-control" by the operator himself, in which skills and responsibilities are assumed by himself/herself [2], but not objectified by an external point of view. Each member is a part of the system and they have to work toward the same goal for the institution to reach its goals and insure the quality of care [19].

However, this verification dimension, in order to ensure the quality of the tasks performed, is quite crucial when it comes to such sensitive data as those referred to in this article: respect for medical confidentiality, quality of transmission of information, management of the risks of records leakage, etc. Taking leadership within the MCHM then seems to represent an additional element in the performance of strategies to protect patient medical records. The leaders act here as an element of internal data protection control, which compliance must be assessed and objectified externally by a notational Data Protection Officer [2]. This perspective can also be considered by taking as a model the Zacklad cooperative transaction logic reading grid, framework for analyzing action and practice at the *meso* level [16].

VI. General lack of use of the national digital health record

The MCHM tends to present organizational processes in accordance with public health requirements, while following effective procedures for the management of patients' personal records. It should, however, be noted that the national digital Electronic Health Record (in French, *Dossier Medical Partagé - DMP*) is not integrated into any of the care management approaches within this institution. The study case tends to reveal an attitude of rejection of this instrument by MCHM staff members. They themselves state that this folder "is not designed for medical practice", although they admit the promising nature of such a tool.

"In case of emergency, the DMP becomes counterproductive. Of course, all the information about the patient is included, but it is not sorted or classified. It is up to us to find the right information and, in emergency situations, we have something more important to do than sorting information" says E., one of the MCHM doctors. They also confess that they are disturbed by the additional and time-consuming actions required to update the patients' DMP, since this platform does not provide any automatic downloading add-ons for the software they use. It seems that the DMP system is in contradiction with the practices of MCHM professionals, with regard to their quality management processes. This notion is closely linked to the need for procedural rationality of the care action, namely an "orientation of the activity". In this orientation, the action is justified by taking into account the way in which the tools contribute to performance. It is partly defined by the quality of the realization process [17]. Doctors highlight a logic "inherent in our relationship with objects and our environment, which we judge according to their adaptation to our expectations and needs" [2]. Applying this to Doucet reasoning, it would seem that doctors judge the use of the DMP in terms of its field operability and its ability to meet the needs imposed by their profession.

However, according to their statements, data's quality processing on the DMP is incompatible with their needs. Charlotte Maday, in her article [17], uses the image of deep-sea fishing: throwing a net on the ocean floor, collecting information indiscriminately and presenting it to users. This image seems to be applicable to doctors' feeling towards the DMP. By presenting the "raw" data, the system is not in line with their requirements for efficient data management. It does not fit their ethical approaches or the "informational lean", the main data management strategies used within the MCHM. More broadly, DMP raises the need for co-production concerning innovation and the necessary collaboration between producers and users to guarantee the quality of a product or a service. Bringing together the documents and data, for process governance, requires mastering the notion of a system but, above all, acting in a spirit of active collaboration [17]. The main issues related to the DMP concern its digital features, ethical uses of data and the ability of professionals from various trades to collaborate on the same project. It represents a question to investigate, specially concerning the conception and the interoperability of various medical software. The goal is to ensure the performance of software and improve the quality of care.

VII. Managing Teams and Data Security in Times of Crisis: an hypertelic context

The MCHM was severely damaged by the COVID-19 crisis, placing its processes in a position that could be described as "hypertelic". This concept, developed by Simondon, originally applied to technical objects, particularly industrial objects [6]. It develops the idea that a tool specifically developed for a technology or for a specific context becomes obsolete and inadequate if the technology or context evolves. The hyper-specialization of tools or, in other words, a marked hyper-adaptation between a context and an object, can therefore lead to the "de-adaptation" if the environment evolves.

In the pre-COVID-19 context, the public health situation was relatively stable, which served as benchmarks for the development of the organizational processes of the MCHM: identification of very specific problems, seasonal epidemics management, stabilized external collaborations, rigor in data processing (technical and human), etc. The pandemic shattered these established paradigms as quickly as it did drastically. It put the institution's organizational processes to the test, particularly with regard to the protection of patients and data. There was an increasing risk of stress-induced errors, unintentional disclosure or a danger of a breakdown in organizational trust, which is a fundamental pillar of the organization of the MCHM. The pandemic highlighted the need for the institution to adapt to changes in its usual health context. It also highlighted its difficulty to do so. This difficult context contributed to the loss of adequacy of the organizational and operational processes of the MCHM. It led the organization to readapt itself profoundly and urgently, in less than a week. The crisis led to the emergence of what could be described as an organizational hypertelia, a "mismatch" of organizational processes with the context in which they are inserted.

This context was completed by the lack of reference points that could help cope this unprecedented situation. The goal for the MCHM was twofold. The first was to maintain the level of quality of care and the second was to protect the identity of infected patients. In a climate of anxiety and marked suspicion, it was important for the organization to guarantee the safety of its patients and its staff. Tracking infected patients without disclosing confidential information does not only required organizational adjustments but also a significant amount of data management work. The bulk of the work was not so much based on the technical aspects of data storage as on the human and operational aspects. Human and operational issues were identified by the doctors as the two main objects mobilized to deal with the crisis.

VIII. COVID-19: THE NECESSARY EVOLUTION OF ORGANIZATIONAL MANAGEMENT

Within the MCHM, the crisis situation would have led to a certain degree of organizational fragmentation, resulting in a lack of team cohesion, a loss of coherence in decision-making, and difficulties in establishing an action plan or prioritizing. There was also a high risk of a loss of organizational confidence and a backlash that could have put the organization in a position of failure. It seemed relevant, in this context, to appeal to the four organizational issues in times of crisis, defined by Le Cardinal [18]: risk, uncertainty, complexity and sense of values. The risks, especially before the peak of the epidemic, were multiple, whether individual, organizational or health-related. This leads to the notion of uncertainty. During the survey, MCHM practitioners said that none of them, including the most experienced, had ever, in their professional or personal life, faced such an emergency situation: "No matter how much I think about it, I can't find anything comparable to what is happening to us now. How should we deal with a pandemic whose magnitude and consequences stay unclear?". The situation was particularly complex for the organization, as it required a hasty and profound redesign of its operations, while at the same time ensuring data and patient protection. From an operational point of view, while the substance of the work remained the same, the health necessity induced a change of working habits to which the staff had to adapt quickly. From a procedural point of view, the disinfection measures proved difficult to reconcile with the necessary pace of consultations. From a relational point of view, the staff had to manage a multitude of strong emotions to ensure the smooth running of the MCHM, which could have had quite detrimental effects on the cohesion of the group and the quality of service. Cohesion is an indispensable element for the solidity and the security of an organization. This brings us to the last point made by Le Cardinal, namely the sense of values [18]. While the first three points have sometimes been thorny or grueling for the MCHM staff, this notion of value has proven to be rich and redeeming, helping to strengthen the staff's confidence in the organization.

A. Trust as an organizational key resource

One of the main challenges was to protect confidential data from accidental or deliberate misuses (e.g., find out who is infected to protect oneself and one's families). The team needed to emphasize strong group cohesion and deep trust to limit the perverse effects of fear and uncertainty. Many authors recognize the fundamental nature of trust in the consolidation of professional cooperation relationships [18] 22] [23]. They also insist on the importance of trust for good health of the organization. This notion is all the more important in times of crisis or in risky situations. It contributes to the fluidity of actions, exchanges and collaboration, by minimizing the costs induced by action control procedures [24]. Indeed, within organizations that doubt their own competences, control activities reduce the quality of procedures and decision-making. They also increase process times, not to mention the consequences in relational and psychosocial terms. Trust is all the more important as it plays a central role in times of crisis and in risk exposure, as it partly conditions their management [25] and their consequences.

One of the pillars of the successful crisis management was the revaluation of the values defended by the MCHM: dedication, protocol rigor and competence. The young doctors took over the reorganization and the whole COVID-19 medical consultations. This *esprit de corps* strongly contributed to the reinforcement of positive values within the team (solidarity, team spirit, devotion, commitment, etc.). Moreover, the young doctors not only established the new care protocol, but also expose themselves to the virus, by taking in charge the COVID-19 consultations. In doing so, they diminished the exposure of at-risk doctors to the COVID-19 (age, comorbidity factors, etc.). They took some risk to protect their colleagues. Within the new protocol, the main space of the MCHM was isolated from the space dedicated to COVID-19. The doctor's goal was not only to limit staff exposure to the virus, but also to maintain patient anonymity. The aim was to protect the personal records of potential infected patients and to subtly inform the rest of the staff of the limits not to be crossed.

The dedication of doctors to the organization of care and the protection of staff and patients was a decisive factor. It has a strong impact on the consolidation of trust during the riskiest moments of the crisis, according to the survey results. A hundred percent of the respondents agree that this was the event that marked them the most, that gave them enough confidence to stay united and that gave them a frame to follow for the rest of the crisis. This feeling has to do with the relationship between individual focus and collective focus, between profit and risk. By taking the risk of exposing themselves to protect their colleagues, the young doctors valued the notion of dedication and the confidence they had in their team's ability to deal with the crisis. Moreover, their attitude towards the protection of their infected patients' data was a reminder of the ethical conduct they expected from all the staff.

B. Emotional intelligence: a pillar of crisis management

The doctors expressed an esprit de corps, a withdrawal from individual benefit for the benefit of the collective, of the whole organization. This strong decision helped to strengthen the commitment to the collective and clearly defined their role within the organization [2]. At this worrying and uncertain moment of the crisis, their involvement represented an organizational anchor point, a reference point for all staff, enabling them to become aware of the values at work in their work [19; 20]. From a meso point of view, this leads to a pronounced form of professional and emotional commitment of all staff. The doctors' action was both a translation and a reminder: it reflected the ethical conduct expected during the crisis, while reminding staff that they could trust the rigor of the organization. This attitude echoes to a communication studies approach, which considers emotions as a social performity [26]. According to this approach, emotions and affectivity are not considered as individual construction, but as a product of social interactions, forget by the socio-cultural culture of the organization [26]. As the organization benefited from a long-term and strong affective and emotional bonding, it seemed evident to capitalize on it to manage the first days of the crisis.

The stakes of this event were both rational and emotional. As Goleman reminds us, "emotional intelligence is based on self-control, ardor, perseverance and the ability to incite oneself to action" [19] [20]. It is based on fundamental ethical attitudes. The rational mind, of which we are most aware, is balanced and reflective, while the emotional mind is impulsive, powerful and sometimes illogical. The attitude of doctors has appealed to this dimension of the mind, acting against the deleterious effects of agitation, persuading - not convincing - the rest of the team to act to achieve a common goal. They acted on feelings rather than on reasoning, on the capacity for commitment and influence.

Rational and emotional intelligence are inseparable from each other for the well-being of organizations. Alvarez, in 2001, highlighted the constant need to combine the rules and procedures of formal systems with the emotional dimensions of the team. These are plural and circular relationships [27]. Within the MCHM, every action taken, every protocol established acts on the psychosocial determinants of the team. The use of masks and gloves, but above all of hats and gowns, which were not mandatory, had the effect of reflecting the organization's desire to protect its employees. The same is true of the protocols for receiving COVID-19 patients, as well as the systematic disinfection processes, which were intended to enhance staff protection. By acting on the operational and procedural level, the MCHM was also able to act on the affective dimensions of its team and enhance its interest in them.

The collective action can be related to the ability of the organization to be resilient, e.g., to “absorb, respond to and also benefit from events that occur as a result of changes in the environment” [28]. The hypertelic context can lead the organization to a definitive failure. It was all the more crucial to reduce its effects during the crisis as such a failure could endanger the whole territorial healthcare network and the population. Resilience rests on a system of shared values and an ability to use owned resources in a creative and innovative way [29]. The MCHM was able to draw on its strengths, particularly its human and procedural strengths, and on its value system to stem the adverse effects of the crisis. As the organization could not benefit from any additional technical and human support, it intelligently reviewed its entire procedures, in order to make the best use of the resources at its disposal and to adapt as best as possible to the management of the crisis.

C. The essential circularity between technical and human aspects

An organization is not only based on these non-human and technical aspects, but on an alliance of these dispositions with the social and human space of which it is composed. Yet one of the staff members showed a contagious anxiety, despite this collective effort to face the situation calmly. It resulted in growing restlessness, errors in the management of patient data, difficulty communicating with others, and proven inefficiency in performing their tasks. Thus, both procedurally and humanly, the employee undermined the organizational stability and security of the MCHM. This was all the more worrisome since the psychosocial stability of the MCHM remained precarious, suspended from daily government announcements on the evolution of the epidemic. The decompensation of the employee risked causing internal dysfunction, the repercussions of which could be disastrous externally. Indeed, each individual is part of the system and each exerts an influence on all the others [19] [20]. The decision to allow this employee to work from home allowed the preservation of other staff members and the quality of service to patients. This example clearly shows that a system or network does not have an immutable durability, but on the contrary must be vigilant to each of its components to ensure their cohesion.

Everything is thus equally important in the organization: organizational factors (procedural capacity), cognitive factors (interpretation by individuals), discursive factors (information exchanges), as well as non-human entities. This whole constitutes the organizational collective, the network. The stability of the network depends on the solidity of the components and their ability to anticipate and cope with future events. It is this anticipation of events that has been the lifeblood of the MCHM. By isolating the disruptive actor before it impacted the rest of the organizational components, they made it possible to safeguard the stability of the network and data security, while offering a benevolent and adapted solution. It is difficult for an organization to find a constant balance, especially when it comes to human components in a crisis situation. The strength and quality of a network is based on its ability to adapt to its social context (Callon, Latour, Akrich, etc.). The social dimension is understood as an effect caused by interactions between very different actors, who are continuously succeeding each other and who participate in the evolution and modification of the network. Within the MCHM, there were successive, progressive and necessary adaptations of the organizational processes, which reinforced its capacity for collective action and fought against its hypertelic dimension.

IX. Organizational Healthcare Network: a model of collaboration and interdisciplinary cooperation

The organizational "readjustment" solution chosen by the MCHM was interdisciplinary cooperation. The beginning of the pandemic was, according to the doctors interviewed, the most complex moment in their crisis management. Starting from scratch is one thing, making this new beginning efficient is another. According to Pesqueux, complexity, uncertainty of outcome and the dangers involved call for cooperation [24]. As soon as the state of emergency was made official, the objective was to collectively find the most satisfactory solution to ensure continuity of care, while avoiding antagonisms, paradoxes and contradictions. This explains the rejection of the action plan proposed by the leaders and the health framework. This, according to one of the young doctors, "was, in itself, not a bad idea. The substance seemed good to me, but there were too many contradictory elements in the procedures. There were also a lot of redundant steps, which wasted unnecessary time and were not relevant to the management of COVID-19 patients. We absolutely had to involve the surrounding health care system. Not only to refine our strategy, but also to ensure consistency. This excerpt illustrates a crucial point in crisis management. In a situation like this, it's not so much a matter of having experts on specific topics. Rather, it is a matter of consolidating a network of actors capable of collaborating and producing collective responses to problematic situations, while remaining united and solid to ensure an efficient and constructive organization [24].

In general, the organizational healthcare network constitutes a tool for improving professional practices and, more broadly, it contributes to maintaining or even improving the level of quality of service provided [10]. The network must act on three levels of collaboration, all of which are essential [5]: the operational (or clinical) level - where the acts of care are carried out -, the structural level - which mobilizes human and material resources to ensure continuous and comprehensive care - and the institutional level - representing the decision-makers and financers of the health system.

With regard to the operational level, the MCHM relied on the interdisciplinary nature of the organizational health network, in particular the hospital institutions. The latter benefit from numerous plans and protocols specific to crisis management, which propose efficient and specific organizational processes to respond to the multiplicity of types of crisis. Thanks to the solidity of their professional network, MCHM doctors were able to quickly obtain the information needed to structure their care protocol. The latter represents a procedural hybridization. On the one hand, it is based on the principle of sectorization in the hospital environment, which was gradually implemented in all institutions. However, since hospitals were better adapted to procedures for isolating contagious patients, the MCHM team had to adapt the procedure to its own structural constraints. This required a collaborative reflection in order to find a solution that would be comfortable for patients, while limiting the risk of exposure for professionals.

The operational cooperation of the MCHM local health network also developed a structural cooperation. As hospitals feared to be overwhelmed by the number of patients, MCHM doctors volunteered to replace hospital doctors during their days off to maintain the quality of care of the surrounding hospitals. Their objective was to maintain the quality of patient care, so that it would be comprehensive and continuous [30]. There seems to be, within this organizational health network, a certain conception of solidarity, which associates cooperation to the realization of a common work [24]. This case is in line with the organizational model known as "collaboration - cooperation" and corresponds to a plural level of relationship, which enriches and fleshes out the interdisciplinary relationships between institutions. This organizational principle tends to reject practices of non-cooperation and domination and instead aims to promote relational exchanges. This inter-organizational alliance was, it seems, a determining factor within the Landes health territory.

X. Conclusion

During the first phase of the survey, the sharing procedures introduced at the MCHM raised questions about risk assessment. In this institution, different informational and managerial strategies were put in place to secure the exchange of personal records. However, during the first investigation, it seemed that the process of quality of care evaluation and “information crisis” management protocols (accidental or fraudulent disclosure, for example) were relatively minor. Before the COVID-19 crisis, the institution did not seem to have been confronted with any type of major crisis. Its youth and its efficient management could explain this situation. Emergency protocols to resolve this type of situation were non-existent in the MCHM. It is problematical, as this implementation is one the most important principles of evaluating the quality of care in France, since the publication of the law of 31 July 1991 [13].

It, therefore, seemed essential to study in great depth the involvement of the MCHM during the crisis and the change it had to conduct to brave it. The COVID-19 had been the real first challenge that could have put the MCHM in danger or even in total failure, especially concerning personal records management. During first month of the pandemic, the MCHM invested in a global reflexive stance to modernize its processes [30]. It questions its own modes of management to adapt to the crisis and so its capacity of resilience, as an ability of the organization to act in adequation to the reality, without taking any unthinking risk [29]. In the field of health, apparently minor errors or failures can have vital consequences [1] or endanger a health care institution and its staff. The management of the quality approach of such an institution cannot be done without a risk management component, nor a more global and formalized evaluation aspect. The measurement principles are inseparable from the principle of quality management [10]. It was necessary for the MCHM to proceed to this modernization and to develop its quality management procedures.

The second part of the survey highlighted the importance of organizational trust and the mobilization of emotional intelligence. Quality and data management rely to some rational perspectives. However, the others aspects of the institution depend on human and emotional aspects, even considering data management. The price for the lack of emotional intelligence or trust could end up compromising the existence of the organization [19] and its efficiency during the crisis. Before the COVID-19 pandemic, problems were resolved with discretion and fluency, with clear and identified leaders. The case of the MCHM presents a very interesting duality. Indeed, despite the profound redesign of these organizational processes, its founding values (commitment, solidarity and trust) had been critical for its success [19] and had insured the stability and the fluency of the MCHM team. The survey could thus conclude to the predominance of trust and emotional intelligence over processes and management of technical devices. It also highlighted the necessity for healthcare institutions to be resilient and to develop a continuous adaption to its environment, to avoid hypertelia, and to learn from its crisis experiences to improve its processes [31].

It would be interesting to investigate, on a larger scale, other MCHs, in order to compare the results of this study with other territorial and technical contexts. Health care institutions were severely tested by the epidemic of the COVID-19 and some of them did not experience the success of the MCHM. Conducting a large-scale survey would test the hypothesis of the preponderance of emotional intelligence in crisis management. It could also help identify the key resources on which organizations should rely during crisis contexts. In a longer-term perspective, the results of such a study could formulating crisis management approaches dedicated to emerging and collaborative health organizations. The objective is also to consider the development of a single working model for all MCHs, specially concerning data privacy management. As each MCH has its own specifications (socioeconomical context, number of the staff members, equipment, competences, etc.) would it be possible and relevant to propose a single crisis management model and resilience capacity to all of them [32]?

REFERENCES

[1] E. Vigouroux-Zugasti, C. Bourret and S. Parrini-Alemanno, “From Theory to Reality – Health Data Management in a Complex System”, eTELEMED 2020, The Twenlfth International Conference on eHealth, Telemedecine, and Social Medicine, DigitalWorld 2020 Congress, November 21, 2020 to November 25, 2020. [Online]: https://www.thinkmind.org/index.php?view=instance&instance=eTELEMED+2020 [Accessed: 10-November-2020].

[2] C. Doucet, Quality, University Press of France (Presses Universitaires de France), 2013.

[3] The French Senate. Access to care: health innovation promoted in French territories, 2017. [Online]: http://www.senat.fr/rap/r16-686/r16-68613.htm [Accessed: 16-January-2020].

[4] C. Vilcot and H. Leclet, “Quality indicators in health: Certification and evaluation of professional practices”, Second release, French Association of normalization, 2006.

[5] A. Bartoli and J. Sebai, “Multidisciplinary Care Houses facing territorial coordination”, Gestion 2000, no. 6, pp. 17-37, 2015.

[6] New Aquitaine Regional Health Authority, Population and its characteristics in New Aquitaine, 2018. [Online]: http://www.nouvelleaquitaine.ars.sante.fr/la-population-et-ses-caracteristiques-en-nouvelle-aquitaine [Retrieved: January, 2020].

[7] INSEE, Complete File – Landes (40), 2010. [Online]: https://www.insee.fr/fr/statistiques/2011101?geo=DEP-40#chiffre-cle-1 [Accessed: 23-October-2019].

[8] New Aquitaine Regional Health Authority, Key Metrics, 2017. [Online]: https://www.calameo.com/read/003713143e21e889f9535?page=3 [Accessed: 23-October-2019].

[9] French Data Protection Act. *Loi informatique et liberté,* Law no. 78-17 of January the 6th 1978 (Last modification: July 27th 2019). [Online]: https://www.cnil.fr/fr/la-loi-informatique-et-libertes#article2 [Accessed: 23-October-2019].

[10] L. Miguel-Chamion, “Protection of shared health data”, International Journal of Bioethics, vol. 25, no. 3, pp. 63-80, 2014.

[11] J. Béranger, “Data ethical value of health big data”, Les Carnets du numérique, vol. 12, no. 1, pp. 109-32, 2016.

[12] C. Vilcot and H. Leclet, “Quality indicators in health: Certification and evaluation of professional practices”, Second release, French Association of normalization, 2006.

[13] L. Monnoyer-Smith, “For a complex epistemology of information and communication sciences”, presented at the 16th SFSIC Congress, Paris. 2008 [Online]: https://www.sfsic.org/congres_2008/spip.php?article78, [Accessed: 23-October-2019].

[14] M. Zacklad, “Management as a scoping activity in a context of cooperative transactions”, Activité, vol. 10, no. 1, pp. 192-200, 2013.

[15] D. Maurel, “Sens-making: A model for the construction of reality and the apprehension of information by individuals and groups”. Communication Studies, vol. 35, no. 2, pp. 31-46, 2010.

[16] G. Le Cardinal, “Federating and stimulating energies, building trust to face crisis situations”, Inflexions, vol. 2, no. 3, pp. 153-168, 2006.

[17] G. Le Cardinal, “Trust as a fundamental element of society”, Revue Projet, vol. 2, no. 293, pp. 65-71, 2006.

[18] D. Goleman, Emotional intelligence, Bantam Books, 1996

[19] D. Goleman, Working with emotional intelligence, Bloomsbury Publishing PLC, 1999.

[20] C. Mayday, “Quality of data and Open data”, I2D Information, données documents, vol. 53, no. 4, pp. 58-58, 2016.

[21] Software Weda. [On line]: https://www.weda.fr [Accessed: 26-March-2020].

[22] D. Maurel and A. Chebbi, “The perception of informational trust. Impacts on informational behaviours and documentary practices in an organizational context”, Communication and Organization, vol. 42, pp. 73-90, 2012.

[23] G. Simondon, On the Mode of Existence of Technical Objects, (ed. Aubier, 1958) University of Minnesota Press, 2017.

[24] P. Peretti-Watel, Risk society, La Decouverte, 2010.

[25] F. Alavrez, The role of trust in the exchange of information: Study of control relationships in the hospital environment. Human Sciences and Society, Parix IX Dauphine University, 2001. [On line]: https://hal.archives-ouvertes.fr/tel-02545337/document [Accessed: 26-July-2020].

[26] C. Lengnick-Hall and T. Beck, “Adaptive fit versus robust transformation: how organizations respond to environmental change”, Journal of Management, vol. 31, no. 5, pp. 738-757, 2005.

[27] A. Giddens, The Consequences of Modernity, Standford University Press, 1990.

[28] L. Begin and D. Chabaud, “Organizational resilience: the case of a family business”, RFSIC, no. 200, pp. 127-142, 2010.

[29] L.D. Coutu, “How resilience works”, Harvard Business Review, vol. 80, no. 5, pp. 46-55, 2002.

[30] A. Dumas and F. Martin-Juchat, “A communicational approach to emotions in organizations: questions and

methodological implications", RFSIC, no. 9, 2016. [On line]: http://journals.openedition.org/rfsic/2103 [Accessed: 26-July-2020].

[31] M.K. Christianson, Farkas M.T., K.M. Sutcliffe, K.E. Weick, "Learning through rare events: significant interruptions at the Baltimore & Ohio Railroad Museum", Organization Science, vol. 20, no. 5, pp. 846-860, 2009.

[32] J. Bauweraerts and O. Colot, "Organizational resilience in family business: myth or reality?", Recherches en Sciences de Gestion, vol. 2, no. 101, pp. 197-215, 2014.

Principles and Applications of a Multiresolution Geodata Cube

Lassi Lehto, Jaakko Kähkönen and Juha Oksanen
Geoinformatics and Cartography
Finnish Geospatial Research Institute (FGI)
National Land Survey of Finland
Masala, Finland
email: lassi.lehto@nls.fi, jaakko.kahkonen@nls.fi, juha.oksanen@nls.fi

Abstract—**The Data Cube concept provides a useful metaphor for management of raster geodata resources in the cloud. An initiative, called GeoCubes Finland, has been launched with the aim to facilitate access to geospatial raster data for academic research. The work is carried out in the context of a major research infrastructure development program in Finland. In the ingestion process, data sets are pre-processed into a harmonized, multiresolution cloud-based repository and brought into a common georeferencing frame, resolution levels, encoding format and tiling scheme. A custom Application Programming Interface (API) has been developed for flexible query, download and analysis of the repository's content layers. Cloud-optimized access to pre-stored resolution levels supports efficient interactive visual exploration of analysis results computed on-the-fly. The presented application examples of the GeoCubes repository include field inspection, route finding and wind exposure analysis.**

Keywords-raster data; multi-resolution; harmonisation; cloud service; visualization.

I. Introduction

Use of raster-formatted data sets in geospatial analysis is increasing rapidly. At the same time geographic data are being introduced into disciplines outside the traditional domain of geoinformatics, like climate change, intelligent transport, health care and immigration studies. Geospatial data sets are also growing rapidly in number and volume. In particular, the volume of raster data sets is becoming difficult to manage. The resolution of image sensors has steadily improved and new imaging technologies have been taken into use. In addition, integrating geospatial raster data sets for analysis has become a tedious task, as the data sets typically differ in critical parameters like origin, resolution, coordinate reference system, encoding, format, etc. It is important to develop mechanisms that facilitate access to relevant data resources.

The data cube concept has emerged as a solution for organizing a repository of harmonized geospatial raster datasets [1] [2], [3]. In computer technology, a data cube is a multi-dimensional array of values [4]. Those values represent certain facts that can be organized along various aspects. For instance, results of a vote can be considered along political party, voting districts, age or gender of candidates, year of election, etc. These aspects correspond to the axes of the multi-dimensional array and the values live in cells inside this array.

In the geospatial domain, the data cube approach was first used in the Earth Observation (EO) community for organizing vast amounts of satellite images [5]. In this application area time is a very important dimension, as satellite imagery are typically available in extensive time series. The four most typical data cube dimensions in an EO application thus are the two geospatial coordinate axes, time and the imagery type.

In the case of a geospatial data cube, the predominant axes naturally are latitude, longitude and the possible height. A voxel-based approach for organizing truly 3D geodata is also possible [6]. The cell value represents the fact about the physical environment that the data set happens to describe. Thus, content theme can be seen as a predominant axis of a geospatial data cube. The most important benefit of organizing data sets as a data cube repository is immediate availability of the contents for integrated analysis. The approach thus aims at fulfilling the goals of the concept Analysis Ready Data (ARD) [7].

The most important positive aspects that a multidimensional data cube approach provides for geospatial application can be listed as

- A data cube forms an integrated data repository, where harmonized content layers are ready for use without troublesome data preparation steps.
- Ingested datasets are aligned on pixel level and thus can be readily integrated for visualization and analysis.
- A data cube can be accessed along any of its dimensions, allowing for new kinds of knowledge retrieval and spatial analysis processes.
- A joint data cube repository enables data providers to deliver content through a new, easy-to-use channel, thus expanding their user base.
- The simplicity of raster data processing can be more effectively and widely exploited.
- By organizing a data cube as a cloud-based service with Web-friendly APIs, the potential use scenarios can be further widened.

The importance of the data cube concept for geospatial application domain is also demonstrated by the fact that the Open Geospatial Consortium has initiated standardization work on the subject [8]. Another significant development is the Open Data Cube (ODC) initiative. ODC is an open source software library for organizing vast amounts of satellite imagery according to principles of data cube [9]. The software includes components for cataloguing and indexing EO resources and for ingesting those resources into a harmonized and optimized storage format for easy data retrieval. Furthermore, the ODC library provides tools for accessing and downloading content and for performing analysis operations on it. The ODC community has also developed tools for visual exploration and statistical analysis of the data cube contents. Concrete national and regional data cube implementations based on ODC include the Australian data cube called Digital Earth Australia [10], the Swiss Data Cube [11], Columbian Cube, Vietnam Open Data Cube, and the Africa Regional Data Cube covering Kenya, Senegal, Sierra Leone, Ghana and Tanzania [12].

ODC can also be applied to data resources other than EO data. The examples mentioned in the ODC documentation include gridded data sets like elevation models, geophysical grids and other interpolated surfaces. However, most of the activities around geospatial data cubes focus on satellite image processing, in general, and on their time series applications, in particular [5].

An initiative has been launched in Finland to build a multi-resolution, cloud service-based geodata cube, called GeoCubes Finland, containing some of the most important national geodatasets [13]. The contents of the GeoCubes include data layers like Digital Elevation Model (DEM), administrative areas, land use, surface deposits and various attributes of the national forest inventory. The content layers are provided by the governmental agencies that collect and continuously maintain them. GeoCubes Finland thus differs from other geodata-related data cube implementations by not focusing on satellite imagery, but rather on other traditional geodata sets. The GeoCubes development is being carried out in the context of a large research infrastructure development program in Finland, called Open Geospatial Information Infrastructure for Research (oGIIR) [14]. Datasets to the GeoCubes are being provided by the National Land Survey of Finland (NLS), the Finnish Environment Institute (SYKE), the Geological Survey of Finland (GTK), and the National Resources Institute Finland (LUKE).

In Section II, the basic principles of the GeoCubes Finland are presented. Section III details the technical implementation of the GeoCubes data repository. Data ingestion process is described in Section IV. Section V discusses the custom content access API of the GeoCubes Finland data repository in more detail. Section VI presents the two GeoCubes Finland client platforms developed. In Sections VII and VIII, some visualization related considerations and example applications are discussed. Section IX concludes the paper.

II. GeoCubes Finland

A. General

A feature that makes GeoCubes Finland different from other geospatial data cube implementations is its multi-resolution nature. A set of fixed resolution levels (1, 2, 5, 10, 20, 50, 100, 200, 500 and 1000 m) have been selected to store the contained data sets. The resolutions are selected to facilitate processing of analysis operations, typically run on round resolution values, and to enable easy integration with statistical and other auxiliary data sets. Opposite to resolution levels typical for web mapping schemas, such as the powers of two in Google Maps, in GeoCubes Finland the resolution levels are selected to serve the user in analysis tasks. The available set of resolution levels depend on the original accuracy of the source data set. Thus, the finest resolution of most of the GeoCubes Finland's content layers is 10 m. Only some national data sets can be reasonably be represented in 1 m resolution. These include the most accurate digital elevation model and the administrative division data sets.

Resolution levels could possibly be seen as one dimension of a multi-dimensional geodata cube, but each resolution level has an individual range for the cube's coordinate-related axes. Therefore, the resolution levels of the cube must in fact be seen as a set of separate cube instances. Hence, the plural form of the cube's name: GeoCubes Finland.

In addition to industry-standard access interface to coverage data, the Web Coverage Service (WCS) [15], the GeoCubes contents can be accessed via direct file URLs and through the custom-built GeoCubes API.

B. Contents

The contents of GeoCubes Finland include a representative selection of spatial data sets maintained by governmental research organisations in Finland (like SYKE, LUKE and GTK). As reference data, some general-purpose data sets provided by the NLS are also included. Data sets are organised as individual layers of information with common representational properties for easy integration and analysis.

Examples of data sets stored in GeoCubes in the first phase include high-resolution elevation models (from the NLS), land-use layers (from the SYKE), superficial deposits layers (from the GTK) and national forest inventory layers (from the LUKE). The latest additions to the GeoCubes Finland data layers are slope and aspect of the terrain, computed from the 2 m resolution DEM, sea regions and the cropland field parcels layer.

C. Metadata

Metadata concerning the data sets stored in GeoCubes Finland are provided as a centralised resource. Because of the nature of the data sets, special attention is put on providing descriptive information about the classifications applied in raster layers. This information is predominantly made available as internal metadata fields of the raster data file. The access API developed to the GeoCubes supports metadata queries, too. As GeoCubes provides multi-resolution data storage, the applied nomenclature in some cases form hierarchical classification structures.

D. Data Encoding

The encoding of GeoCubes Finland cell values depends on the nature of the data set being represented. Both classified data sets (like land use or superficial deposits maps) and data sets with continuous value ranges (like Digital Elevation Models (DEMs) or orthophotos) are included. No-data areas are represented as zero-valued cells in classified data sets and by a separate mask channel in data sets with continuous value ranges.

E. Grid

The standardised grid applied in GeoCubes is based on the Finnish national Coordinate Reference System (CRS) ETRS-TM35FIN (EPSG code 3067). This projected CRS is compatible with the pan-European ETRS89 system. ETRS-TM35FIN covers the whole country in one projection zone and has the false easting value of 500 000 m on its central meridian at 27°E longitude. The origin of the GeoCubes Finland's grid (top-left corner) is located at the coordinate point (0, 7800000). The easting value of the origin is selected to avoid negative coordinates. The northing coordinate value is selected as a round 100 km value, allowing for good coverage of the country.

III. Technical Implementation of the Repository

GeoCubes Finland's data storage is implemented as a set of Cloud Optimized GeoTIFF (COG) files [16], [17]. These files are stored on a cloud service platform, organized in directories by data provider, data set and the edition of data set. The area of the country is divided into tessellation of sixty 100 km * 100 km sized blocks, each maintained as a separate GeoTIFF file to ease processing tasks. The tessellation also facilitates parallelization of various processing steps and works as a rudimentary spatial index. All the files can always be accessed in a straightforward manner via http (HyperText Transfer Protocol) by their URLs (Uniform Resource Locator). The block tessellation is shown in Figure 1.

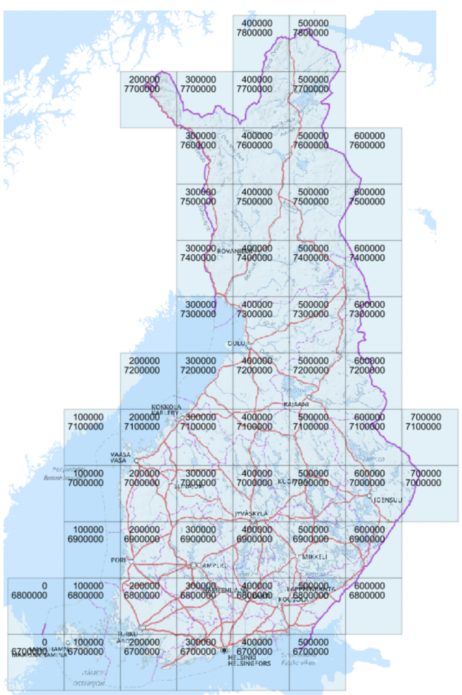


Figure 1. The sixty GeoCubes Finland blocks covering the country and coordinates of their upper-left corner in the national ETRS-TM35FIN coordinate reference system.

A GeoTIFF file arranged according to COG specification contains overviews and has its raster content organized as tiles. The overviews are ordered in sections from the coarsest to the most detailed, followed by the full resolution raster. The metadata describing the offset of each section must be at the beginning of the file. With this file structure, a calling COG-aware client can make use of the http 'GET Range' query, which enables an indicated subsection of a file to be requested. By first requesting the metadata portion, the client can then select an appropriate range of bytes to only download the needed geospatial area of an appropriate resolution, i.e. overview level. This mechanism significantly improves the efficiency of raster data retrieval for cloud-service based applications. In the rudimentary tests carried out on the GeoCubes Finland COG files, it has been noticed that speed improvements of up to 80 % can be achieved with Cloud Optimized files when compared to traditional GeoTIFF files.

The multiple resolutions are implemented in two ways, both as internal GeoTIFF overview layers and as individual external resolution specific GeoTIFF files. Internal overviews facilitate easy content delivery, as all resolutions are contained inside as single file. Separate resolution specific GeoTIFF files in turn enable better control of resolutions for computations run in external applications. The overview functionality, like most other computing tasks in GeoCubes Finland, is implemented using Geospatial Data Abstraction Library (GDAL) [18]. Python-based processing tasks are performed making use of the Rasterio library [19].

To collect the individual files as a complete view covering the whole country, the so-called Virtual Format mechanism (VRT) of GDAL is used [20]. In this approach, an XML-formatted text file is used to refer to the set of image files that constitute the integrated view. As such, a single VRT file can for instance refer to all external GeoTIFF overview files on certain resolution level, thus facilitating analysis and visualization of larger areas. In the same way, a VRT file can refer to various resolution-specific files in a certain tessellation block, providing a light-weight representation of the multi-resolution data set of that block. VRT files can be easily transported across network connections. Because the file references are stored as absolute addresses, the VRT file can be opened in a third-party application. This approach limits downloading of raster content to the spatial area and to the resolution level the user actually needs.

IV. Data Ingestion Processing

In the data ingestion process the data sets being imported to GeoCubes Finland repository are transformed to the common form. Vector data sets are first rasterized. In the rasterization process the value of attribute selected for GeoCubes is stored to the cells. The spatial resolution applied in the rasterization is selected from the list of GeoCubes resolutions, based on the accuracy of the data set in question. All of the data sets currently available in the GeoCubes repository have base resolution of either 1 m or 10 m. In the rasterization process the resulting raster data is stored as GeoTIFF files in the block structure of GeoCubes. If vector data is available in several scales, the smaller scale versions are rasterized and used as the source for overview layers on appropriate resolutions.

If the source data set is available as raster data, the data set is processed to resample the cells to the GeoCubes grid and produce the block-wise GeoTIFF files. In some cases, the classification schemes or data collection resolutions have changed along time between data set editions. This kind of discrepancies must be corrected in the data ingestion process to produce a harmonized collection of repository layers.

The next step in the data ingestion process is to produce the coarser resolution representations as GeoTIFF overview layers. The generalization procedure available in the GDAL's 'BuildOverviews' function is used in the processing of coarser resolutions. An appropriate resampling method is selected, depending on the nature of the source data set. In the process both the resolution levels are produced both as internal GeoTIFF overview layers and as independent single-resolution GeoTIFF files. The multiresolution files are then transformed into the COG form. For categorical data sets the metadata, color table and category information are written to the GeoTIFF files.

In case of categorized data sets, generalization is a delicate process, in which the distribution of categories should be maintained as accurately as possible. If the source data set provides a hierarchical classification scheme, more general classes are used on the coarser resolution levels. In some cases, the generalization procedure has been carried out by the data provider using custom data-specific methods and the resulting multiresolution GeoTIFF files are then used as the input for the GeoCubes data ingestion process.

Next the VRT files are processed. The multiresolution versions of the block-wise GeoTIFF files are combined together in a VRT file. The single resolution files on all available resolution levels are processed in the same way. In addition, for each block a set of VRT files is created, so that for each available resolution level a VRT file is available containing this base resolution and all the coarser resolutions. This arrangement facilitates dynamic provision of multiresolution representations of the content in various base resolutions. Finally, these block-wise VRT files are combined together as VRT files covering the whole country.

Data ingestion procedure is automated by Python script-based processing. GDAL's Python API is used extensively [21]. The main spatial functions like vector data access, rasterization, raster data generalization, georeferencing and GeoTIFF manipulation are based on the use of GDAL Python API. All the processing of raster data arrays is done using Python's NumPy library [22]. The fully automated ingestion process facilitates processing of new data set editions as they become available.

V. Content Access API

An API has been developed for exploring, downloading and analyzing the GeoCubes Finland contents [23]. The API is designed according to the principles of RESTful Web API [24], in which the request is defined by the path components of the query URL. The API provides requests for querying the basic metadata of the repository, querying cell values on a given location, downloading content by bounding box, polygon or administrative unit and analyzing the content in terms of cell value distribution, change detection, etc. The analysis operations are run on the server platform, without downloading any content to the client side.

In all content queries, the query can be run on a desired resolution level and year. The multi-resolution structure of the GeoCubes repository gives to the user a flexible choice on speed vs. accuracy of the operation. For instance, an analysis procedure under development can be first tested on a coarse resolution level, and the real, time-consuming run on the finest resolution be carried out only when seemed appropriate. In case of analysis based on visual exploration, the resolution level, on which the analysis is run, can always be matched with the zoom level of the visualization. This way the analysis can be run in roughly constant time and the interactivity level of the application can be kept stable.

The general scheme of the access API can be described as follows:

```
what to do /
   on which resolution level /
      with which content layer /
         where /
            when /
               how
```

As an example, the query for the cell value on a given location becomes as:

```
legend/500/mvmi-paatyyppi/340500,6695000/2009
```

where mvmi-paatyyppi (in Finnish) is one of the themes (forest inventory data) of GeoGubes.

Another query would request download of a content layer inside the given list of municipalities ('kuntajako' in Finnish), from the given resolution level and year:

```
clip/100/corine/kuntajako:734,761,834,433,224
,444,927/2000
```

The resulting data set, requested using the GeoCubes Web client, is shown in Figure 2.

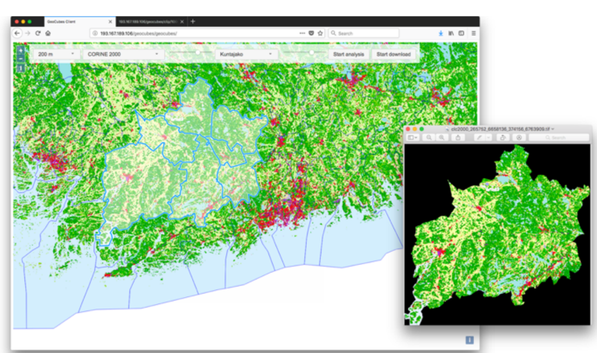


Figure 2. A CORINE data set downloaded via GeoCubes API using administrative units' boundaries as selection area.

Downloading of a DEM from two blocks in VRT format with multiresolution content would go as:

```
clip/20/km10/blocks:300000,6900000,
300000,6800000/2018/vrt:mr
```

Download of a terrain slope layer, computed from the 2 m resolution DEM, in horizontal resolution of 1 m, from inside a user-defined polygon would be initiated like this:

```
clip/1/km2slope/polygon:358978,6668706,358918
,6668645,358923,6668597,358980,6668549,359093
,6668579,359096,6668668,359017,6668719,358978
,6668706/2019
```

An example of an analysis would be change detection between CORINE versions 2000 and 2012 in land use type fields ('pellot' in Finnish) inside the given bounding box:

```
analyse:changedetect/10/corine:Pellot
/bbox:225700,6660000,494300,6740000/2000,2012
```

The result of the above analysis is an image depicting the changes in red (removals) and green (additions).

Another analysis would determine the distribution of cell values inside the two indicated counties ('maakuntajako' in Finnish):

```
analyse:distribution/500/mvmi-
maaluokka/maakuntajako:04,06/2015
```

The result is a list of existing cell values together with their frequencies. The analysis results visualized in the GeoCubes Web client using the D3.js library [25] is shown in Figure 3.

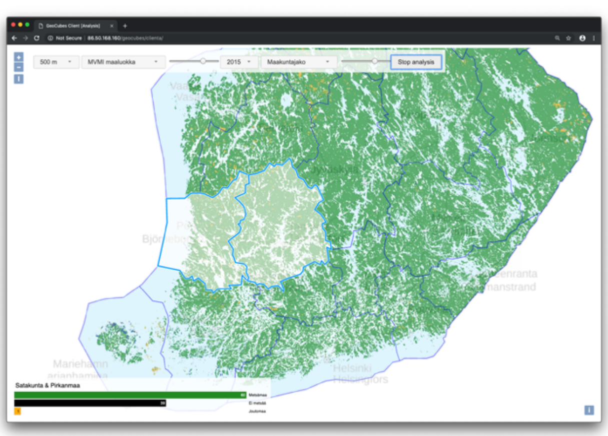


Figure 3. GeoCubes analysis results indicate distribution of data values inside the selected area (shown as a bar chart in the left bottom corner).

As an auxiliary data source for analysis purposes, the Finnish administrative areas in four different levels are also available in vector form in the GeoCubes repository.

The GeoCubes API has been implemented as a Web service using Django Web framework [26], together with Python-based service side scripts making an extensive use of the Python API of GDAL. In addition to the custom API, the most relevant service interfaces standardized by the Open Geospatial Consortium are also supported. The main components of the GeoCubes Finland platform are shown in Figure 4.

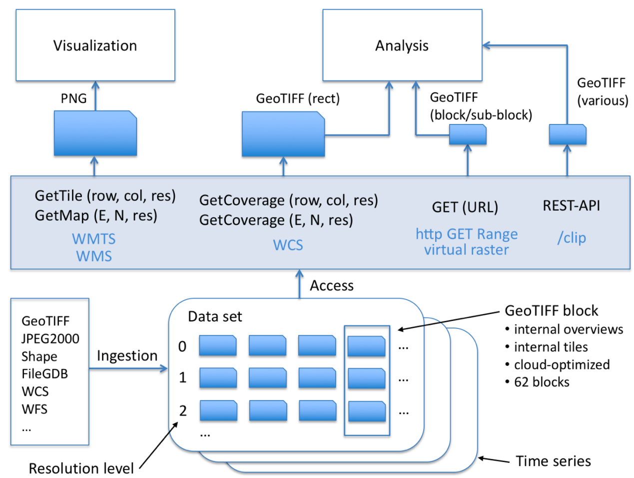


Figure 4. The main system components of the GeoCubes Finland platform.

VI. CLIENT PLATFORMS

Two main software projects have been accomplished to develop client platforms for supporting the GeoCubes data usage. An Open Layers -based web client has been developed to demonstrate the data contents and access API of the GeoCubes repository. The application provides fast and easy visualization of the GeoCubes content layers, together with interactive querying of the layer cell values. The data retrieval options include download by bounding box, user-defined polygon, administrative area and GeoCubes tessellation block. The user interface of the web client is depicted in Figure 3.

A client application capable of accessing GeoCubes API has also been developed as a QGIS plugin [27]. The plugin provides an exploration tool for interactive visualization and querying of the GeoCubes data layers. Data download can be requested using a bounding box, polygon or administrative unit-based selection. The resolution level can be selected explicitly. The downloaded data set can be directly inserted as a new layer into the current QGIS project, or it can be stored to the local disc for later use. An example of the user interface of the plugin is presented in Figure 5.

Figure 5. The user interface of the QGIS GeoCubes Finland plugin.

VII. Visualization Considerations

For basic visualization of the content layers, a Web Map Tile Service (WMTS) is available on the GeoCubes platform. The service makes use of a Web Map Service (WMS) that uses a VRT file combining the original GeoTIFF files as its source. This arrangement enables both ad hoc visualizations through WMS and cache-based static image delivery via WMTS. In the case of the WMTS service, the pre-rendered tiles are available on the fixed resolution levels of the GeoCubes. For best visual presentation, the client should apply scale levels that correspond to those resolutions. The WMS implementation of GeoCubes platform is based on MapServer and the WMTS service on MapProxy.

Visualization of administrative areas poses a specific challenge. Because areas are represented as raster data to facilitate analysis with other content layers, boundary lines cannot be used to separate an area from the neighboring areas. The only way to reliably display the administrative division of the country in a legible manner is to ensure that neighboring areas are always presented with clearly distinguishable colors. The Finnish lowest level administrative division has more than 300 municipalities. Thus, to achieve an appropriate data set with cell values corresponding to the real municipality codes, a raster with 16-bit cell values was created. However, for Web browser-based visualization, a 256-colour PNG image is required. A custom software module was developed that creates random color components for a 256-colour RGB color table and applies the colors to municipalities so that the three-color components of a municipality always differ more than 30 units from all of the neighboring areas' color components. The result is a bright and colorful municipality map, in which all areas can be reliably distinguished (Figure 6).

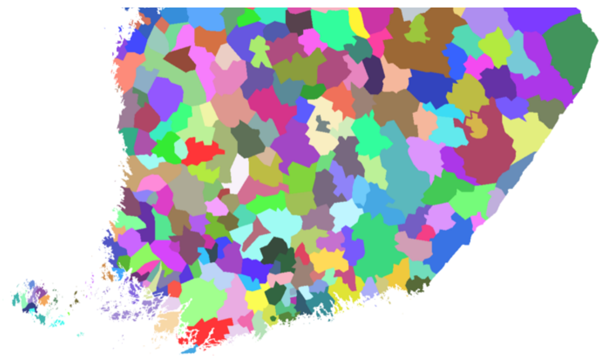


Figure 6. A municipality map with well distinguishable colors.

Another example of enhanced visualization tested in GeoCubes Finland is the locally stretched DEM visualization. The terrain of the country is mostly flat, higher elevations existing only in the northern Lapland. A consistent, stable country-wide DEM visualization thus depicts flat areas with very limited color scale, making it impossible to recognize subtle height differences. A dynamic visualization module was developed for GeoCubes that stretches every individual view requested by the client to full color scale. As the GeoCubes data repository has multiple resolution levels, the visualization processing can be performed efficiently in constant time, independently of the presentation scale of the client. The two images in Figure 7 depict the difference between fixed and dynamic DEM visualization on a particularly flat area in the Finnish Ostrobothnia.

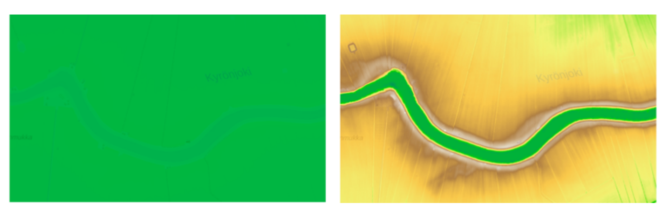


Figure 7. A static DEM visualization compared with a dynamic, locally stretched graphic scale.

A 3D example of visualization of GeoCubes' contents is depicted in Figure 8. Here, DEM data is extracted from the GeoCubes data repository around possible crater locations, where a meteorite impact is deemed to be behind the peculiar rounded land form. The multiresolution contents of the GeoCubes repository supports effective 3D visualization of crater areas of various sizes.

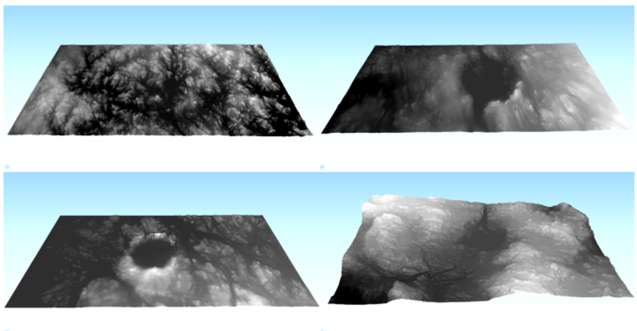


Figure 8. Possible meteorite impact craters visualized, based on GeoCubes DEM. Big scale differences in visualizations are supported by GeoCubes' multiresolution contents.

VIII. Application Examples

The analysis processes can be effectively run on different resolution levels and their results are readily available for visual exploration over the whole range of scales supported by the geodatacube's resolutions. The GeoCubes API has been designed to support access, querying and analysis of the geodatacube's contents on the resolution level appropriate for the actual use situation.

A. *Field Inspection*

As an example of using GeoCubes API, a field inspection of repository cell values has been tested. In this approach, the inspector moves around on the terrain and constantly receives values of the GeoCubes layer of interest into his cell phone. This way, the inspector can compare the environment around him with the stored category values, for instance, he can determine the correctness of forest type information – and do that on all available resolution levels of the repository.

In the developed pilot implementation, the position of the inspector is recorded with an application called Owntracks. The location is reported back to a server (Eclipse Mosquitto) [28] implementing the ISO-standardized Message Queuing Telemetry Transport (MQTT) -protocol. The user is presented with a map-based application (Leaflet with Realtime extension). When the map application gets notice from the MQTT server that the user has moved to a new location, it then send a request to the 'legend' operation of the GeoCubes API together with the location information. As a result, it will get cell values from the requested layer on all available resolution levels and can compare them with the reality around him. The pilot's architecture is presented in Figure 9.

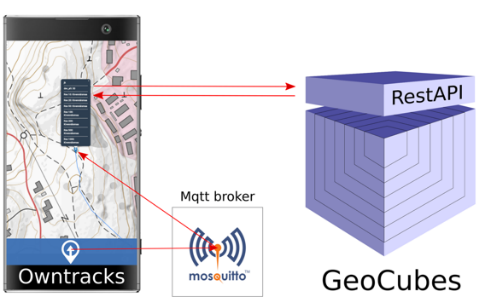


Figure 9. A field inspector accessing GeoCubes cell values using Mosquitto server.

B. *Route Finding*

In another analysis example, the GeoCubes DEM layer is used for route finding for a forest vehicle that has certain limits for movement in steep slopes. DEM data for study area was acquired by accessing the GeoCubes API's 'clip' operation from within the QGIS application [29]. Then, analysis functions available in QGIS were used to compute slope values for the area and these were used as the cost surface for route finding. Areas too steep for the vehicle were excluded from the computation. The case study demonstrates the use of the GeoCubes' content for analysis using tools outside the GeoCubes platform. The same analysis could also be run in different resolution levels, depending on the needs of the application. The resulting route alternatives are depicted in Figure 10 using QGIS plugin [30] for three.js, a JavaScript library for 3D visualization [31].

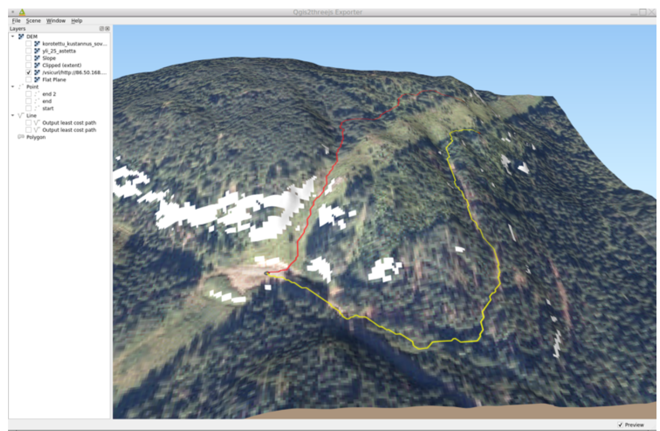


Figure 10. Route finding in QGIS with DEM from GeoCubes using slope as cost surface. White areas denote terrain that has been classified to be too steep and is excluded. The red and yellow lines denote found alternative routes.

C. *Wind Exposure Analysis*

Another GeoCubes data application example involves creating a forecast map of perceived temperature, based on location-dependent exposure to wind. The result is a "winter map" with a background layer, adapted to reflect the snowy conditions of the Nordic winter landscape, together with an overlay indicating the direction of wind and the level of wind chill-affected perceived temperature. The approach makes use of the GeoCubes-provided cloud-optimized multiresolution DEM data, combined with the weather forecast information accessed online from the Finnish Meteorological Institute (FMI).

The service is implemented as a PHP: Hypertext Preprocessor script making an extensive use of GDAL's command line interface. At the core of the service is the GDAL command 'gdal_translate', which accesses the online VRT files of the GeoCubes repository by direct URL reference. The VRT file provides a convenient single point access to all sixty multiresolution GOC files that cover the whole country. Using the 'projwin' and 'outsize' parameters of the 'gdal_translate' command, one can specify from which area the data is to be retrieved and what is the desired size of the resulting image. These parameters together determine the resolution level that is accessed in the GeoCubes multiresolution data storage. This arrangement facilitates an easy and efficient presentation of the analysis results in various scales, depending on the zooming levels selected in the user interface.

Once the request for a "winter map" comes to the service, the PHP script will access the FMI's weather forecast service through a Web Feature Service (WFS) interface, and then parses the required wind direction information from the response. Consequently, the script will request the needed DEM data from the GeoCubes service. The DEM is processed to produce the slope and aspect layers, needed for computing the perceived temperature. After the computation, the resulting wind chill-affected perceived temperature map is rendered. The orthophoto layer is requested from GeoCubes and dynamically tuned to reflect the snowy winter conditions by applying appropriate image filtering tools. Finally, these two map layers are combined and the resulting "winter map" is returned to the client application. An exemplary map, depicting the area of the Levi ski resort in the Northern Finland and visualized for 3D viewing in a three.js-based web client, is shown in Figure 11.

IX. Conclusions

One of the major obstacles for wider use of geospatial raster data sets in different research and analysis scenarios is the work required for pre-processing the available data sets. This often involves coordinate reference system transformations, resampling procedures, coding system translations, integration of map sheet-based data files, etc. To facilitate the introduction of geospatial data sets into research processes in a multidisciplinary setting, a harmonized, easy-to-access data storage would be really beneficial. In the GeoCubes Finland initiative, this kind of approach has been taken.

A representative set of geospatial data sets with national coverage has been ingested into the cloud service-based data repository. The pre-processing phase involves operations like rasterizing vector-formatted source data sets, resampling of source data into the set of standardized GeoCubes resolution levels, harmonization of value coding systems between different data set editions, encoding of the resulting raster content into the common cloud-optimized image format and storing it to the cloud repository, both as binary images files and as textual VRT representations.

By storing the raster data sets in multiple resolution levels, the GeoCubes repository can support certain use cases very efficiently. These include interactive visual exploration of on-the-fly analysis results and testing of an analysis procedure on coarse resolution levels before launching an accurate analysis on detailed levels. The result layer of an analysis can be configured as a new GeoCubes content layer and thus be run dynamically by the calling visualization client. The analysis procedure will always select the resolution level closest to the visual scale, thus enabling nearly constant processing times to be maintained.

Further work on the GeoCubes Finland platform will focus on adding new layers to the repository's data collection, developing modular analysis functions for server-side processing, and enhancing the GeoCubes client applications. More user testing will be carried out to gather feedback and guidance for further development of the platform.

Acknowledgment

We made use of computing resources provided by the Open Geospatial Information Infrastructure for Research (oGIIR, urn:nbn:fi:research-infras-2016072513) funded by the Academy of Finland and CSC – The IT Center for Science Ltd.

References

[1] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, "Harmonized Multiresolution Geodata Cube for Efficient Raster Data Analysis and Visualization," in *GEOProcessing 2020: The Twelfth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain*, Nov 21-25, 2020, pp. 36–41, Accessed: Dec. 7, 2020. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2020_1_60_30055.

[2] P. Baumann, "The Datacube Manifesto," 2017, Accessed Dec. 7, 2020 [Online]. Available: https://external.opengeospatial.org/twiki_public/pub/CoveragesDWG/Datacubes/The-Datacube-Manifesto.pdf.

[3] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, "GeoCubes Finland - A Unified Approach for Managing Multi-resolution Raster Geodata in a National Geospatial Research Infrastructure," in *GEOProcessing 2018: The Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services*, Mar. 2018, pp. 18–22, Accessed: Dec. 7, 2020. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2018_2_10_30069.

[4] Wikipedia, "Data cube," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Data_cube

[5] A. Lewis *et al.*, "Rapid, high-resolution detection of environmental change over continental scales from satellite data – the Earth Observation Data Cube," *International Journal of Digital Earth*, vol. 9, no. 1, pp. 106–111, Jan. 2016, doi: 10.1080/17538947.2015.1111952.

[6] U. Pyysalo and T. Sarjakoski, "Voxel Approach to Landscape Modelling," in *The International Archives of the Photogrammetry and Remote Sensing*, 2008, pp. 563–568, Accessed: Dec. 7, 2020. [Online]. Available: https://www.semanticscholar.org/paper/Voxel-Approach-to-Landscape-Modelling-Pyysalo/a2074c8487c028670348c6efa051d6590a41e6c8.

[7] G. Giuliani *et al.*, "Building an Earth Observations Data Cube: lessons learned from the Swiss Data Cube (SDC) on generating Analysis Ready Data (ARD)," *Big Earth Data*, vol. 1, no. 1–2, pp. 100–117, Dec. 2017, doi: 10.1080/20964471.2017.1398903.

[8] P. Baumann, D. Misev, P. Hogan, R. Brackin, K. Thayer, and E. Hirschorn, "Datacube Domain Working Group Carter," 2017. Accessed Dec. 7, 2020. [Online]. Available:

https://external.opengeospatial.org/twiki_public/pub/CoveragesDWG/Datacubes/17-071_Datacube-DWG_Charter.pdf

[9] ODC, "Open Data Cube," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://www.opendatacube.org/

[10] A. Lewis *et al.*, "The Australian Geoscience Data Cube — Foundations and lessons learned," *Remote Sensing of Environment*, vol. 202, pp. 276–292, Dec. 2017, doi: 10.1016/j.rse.2017.03.015.

[11] SDC, "Swiss Data Cube (SDC) – EO for monitoring the environment of Switzerland in space and time," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://www.swissdatacube.org/

[12] Global Partnership for Sustainable Development Data, "Africa Regional Data Cube," 2020. Accessed Dec. 7, 2020. [Online]. Available: http://www.data4sdgs.org/index.php/initiatives/africa-regional-data-cube

[13] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, "Supporting wide user-base in raster analysis - GeoCubes Finland," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2018, vol. 42, no. 4, pp. 397–402, doi: 10.5194/isprs-archives-XLII-4-329-2018.

[14] OGIIR, "Open Geospatial Information Infrastructure for Research," 2020. Accessed Dec. 7, 2020. [Online]. Available: http://www.geoportti.fi/about/

[15] OGC, "Web Coverage Service," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://www.ogc.org/standards/wcs

[16] OSGeo, "GeoTIFF," 2019. Accessed Dec. 7, 2020. [Online]. Available: https://trac.osgeo.org/geotiff/

[17] COG, "Cloud Optimized GeoTIFF," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://www.cogeo.org/

[18] GDAL, "Geospatial Data Abstraction Library," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://gdal.org/

[19] MapBox, "Rasterio: access to geospatial raster data — rasterio documentation," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://rasterio.readthedocs.io/en/latest/

[20] GDAL, "VRT – GDAL Virtual Format," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://gdal.org/drivers/raster/vrt.html

[21] GDAL, "GDAL Python API," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://gdal.org/python/

[22] NumPy, "NumPy - The fundamental package for scientific computing with Python," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://numpy.org/

[23] L. Lehto, J. Kähkönen, J. Oksanen, and T. Sarjakoski, "Flexible Access to a Harmonised Multi-resolution Raster Geodata Storage in the Cloud," in *GEOProcessing 2019: The Eleventh International Conference on Advanced Geographic Information Systems, Applications, and Services*, Feb. 2019, pp. 26–28, Accessed: Dec. 7, 2020. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2019_2_20_30054.

[24] M. Rouse, "What is a RESTful API (REST API) and How Does it Work?," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://searchapparchitecture.techtarget.com/definition/RESTful-API

[25] M. Bostock, "D3.js - Data-Driven Documents," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://d3js.org/

[26] Django Software Foundation, "The Web framework for perfectionists with deadlines | Django," 2020. Accessed Dec 7, 2020. [Online]. Available: https://www.djangoproject.com/

[27] QGIS, "QGIS Plugins," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://plugins.qgis.org/

[28] R. Light, "Mosquitto: server and client implementation of the MQTT protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017, doi: 10.21105/joss.00265.

[29] QGIS, "QGIS - A Free and Open Source Geographic Information System," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://qgis.org/en/site/

[30] QGIS, "Qgis2threejs Plugin Document — Qgis2threejs Plugin 2.4 documentation," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://qgis2threejs.readthedocs.io/en/docs/index.html

[31] threejs, "three.js – JavaScript 3D library," 2020. Accessed Dec. 7, 2020. [Online]. Available: https://threejs.org/

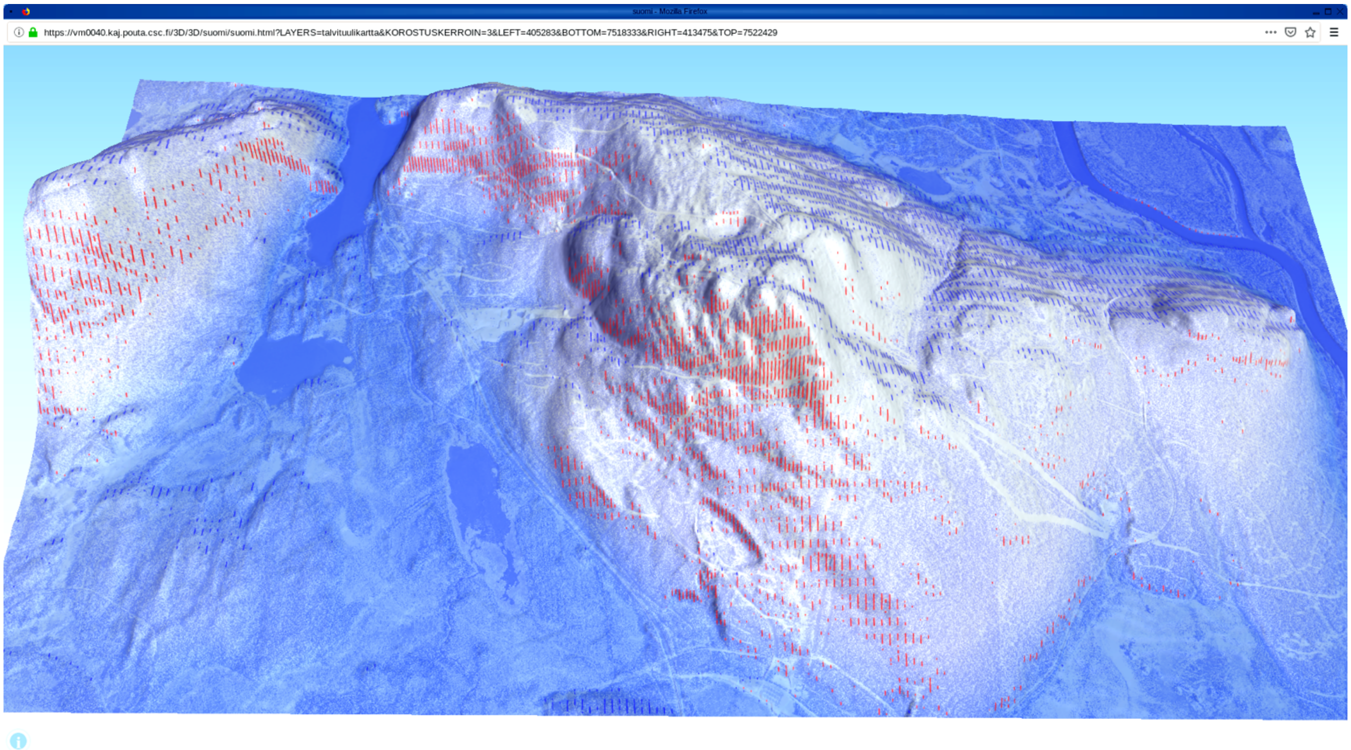


Figure 11. A perceived temperature “winter map” presented for 3D viewing in a three.js-based web client. The areas with red stripes are on the shelter side of the hill, whereas the areas with blue stripes represent chilly slopes, exposed to wind. The orientation of the stripes indicates the direction of the wind and their density the level of the shelter/exposure effect.

Facial Mimicry Analysis Based on 3D Morphable Face Models

Oky Dicky Ardiansyah Prima, Yuta Ono, Hisayoshi Ito
Grad. School of Soft. and Inf. Sci., Iwate Pref. Univ.
Takizawa, Japan
email: {prima, hito}@iwate-pu.ac.jp, g236s001@s.iwate-pu.ac.jp

Takahiro Tomizawa
Hitachi Ind. & Ctrl. Solutions
Yokohama, Japan
email: Takahiro.tomizawa.ax @hitachi.com

Takashi Imabuchi
Office of Regional Collaboration, Iwate Pref. Univ.
Takizawa, Japan
email: t_ima@ipu-office.iwate-pu.ac.jp

Abstract—**Facial mimicry is an important non-verbal communication that can promote favorable social behavior and positive relationships. As recent computer vision technologies have reached the level of human perception for processing facial information, the study of automated analysis of facial mimicry has attracted the attention of the Human Computer Interaction (HCI) society. In this study, we propose a system to evaluate the similarity of facial images based on the shape of the face derived from 3-Dimensional (3D) face data. Two different 3D face data were used in this study: a 3D Digital Character (3DDC) and the Surrey 3D Morphable Face Model (3DMFM). Our approach consists of the following steps: (1) landmark extraction from the facial image; (2) 3D shape fitting; (3) similarity analysis for the face point cloud. Our results show that the similarity between faces can be assessed by analyzing the non-rigid portions of the faces. The proposed system can be extended as a facial mimicry training tool to improve social communication.**

Keywords-mimicry; expression training; emotion; image processing.

I. INTRODUCTION

People often consciously or unconsciously mimic the facial expressions of their conversation partners. This type of non-verbal communication is important in providing additional information beyond verbal communication. This study extends our previous research on a facial mimicry training based on 3-Dimensional (3D) morphable face models [1]. Non-verbal communication refers to communication methods without using languages, such as gestures, facial expressions, tone of voice, eye contact, and posture. Unconscious mimicry increases affiliation, which helps to foster relationships with others [2].

Facial expressions involve signals of a larger communicative process, conveying the focus of attention, intention, motivation, and emotion. For instance, a smile with the corners of the mouth turned up to expose the front teeth may express joy. A frown, typically with the corners of the mouth turned down, forms an expression of disapproval. Some researchers believe that emotions are a universal construct. However, differences in culture can lead to differences in the absolute level of intensity of emotions [3].

The FACS (Facial Action Coding System), which is a set of facial muscle movements corresponding to displayed emotions, is a traditional measure for analyzing facial expressions [3]. The movements of individual facial muscles are encoded by FACS from slightly different instantaneous changes in facial appearance. Each action unit (AU) is described in the FACS manual.

Manual coding of video recordings of participants according to the units of action described by FACS takes a significant amount of time and effort. Automatic analysis of facial expressions has received a great attention from the computer vision community. Bartlett et al. (1999) made an early attempt to automate facial expressions using FACS by applying computer image analysis to classify the basic elements that comprise complex facial movements [4]. Using template matching to detect facial features, such as lips, eyes, brows and cheeks, Tian et al. (2000) developed an Automatic Face Analysis (AFA) system [5], which recognizes changes in facial expression into AUs.

Recent computer vision technology has made it possible to perform conventional face recognition processes such as face detection, face matching, facial feature extraction, and facial expression classification in real time. Baltrušaitis et al. (2018) developed an open source facial behavior toolkit, OpenFace 2.0. This toolkit uses a linear kernel support vector machine to detect facial expressions by detecting the unit of motion (AU) of the face [6]. iMotions [7] is a commercial tool for automatic analysis of facial expressions. The tool allows the user to select the FACET [8] or AFFDEX [9] algorithm for facial expression recognition.

In contrast to the growing interest in the application of automated facial expression analysis, there have been surprisingly few attempts to measure the similarity of facial expressions. In our previous work, we used a deformable 3D Digital Character (3DDC) to find the most similar shape to a given facial image [1]. The similarity of the faces was measured by calculating the relationship between the 3D point clouds obtained from each face.

In this study, we extend our previous work to evaluate the similarity of facial images based on the shape of the face derived from two different face data: a 3DDC [10] and the Surrey 3D Morphable Face Model (3DMFM) [11]. We fit these models to the corresponding facial images and derive the facial features that are important for similarity analysis.

The rest of this paper is organized as follows. Section II discusses known approaches to facial expression imitation. Section III introduces our proposed facial mimicry analysis system. Section IV shows our experiment results. Finally, Section V gives a short conclusion and highlights the most important outcomes of this paper.

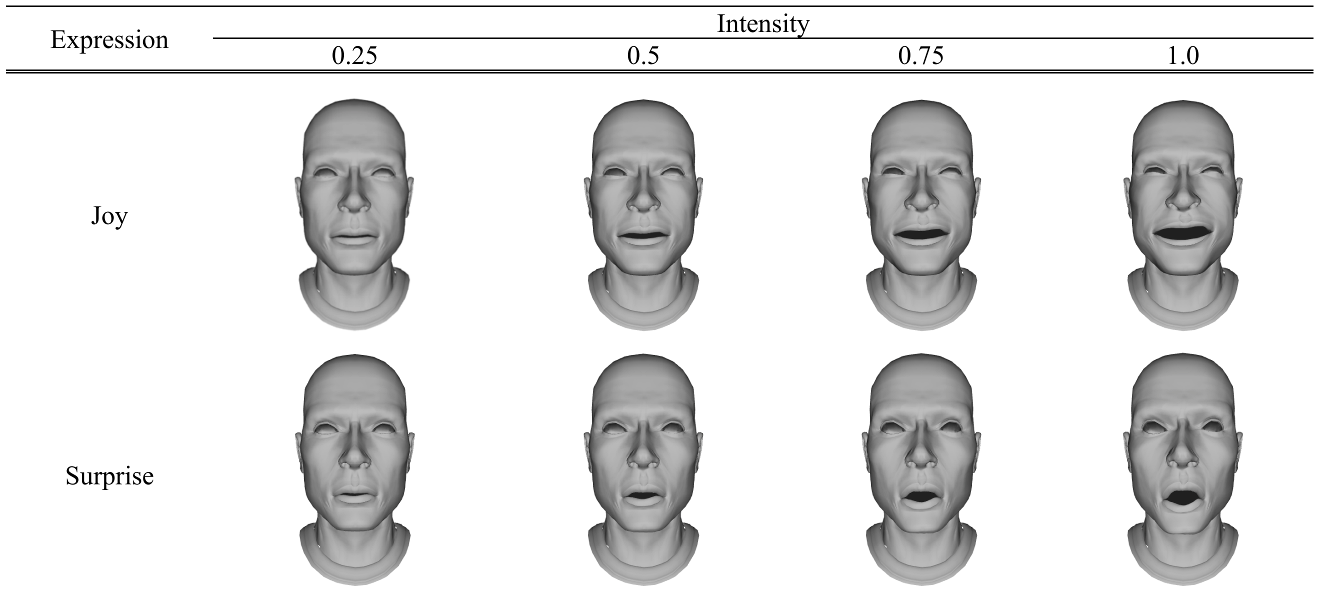


Figure 1. Some facial expressions generated using 3DDC.

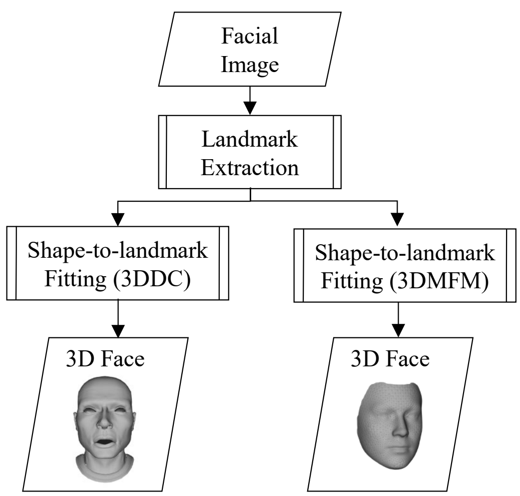


Figure 2. Generation of 3D face data for this study.

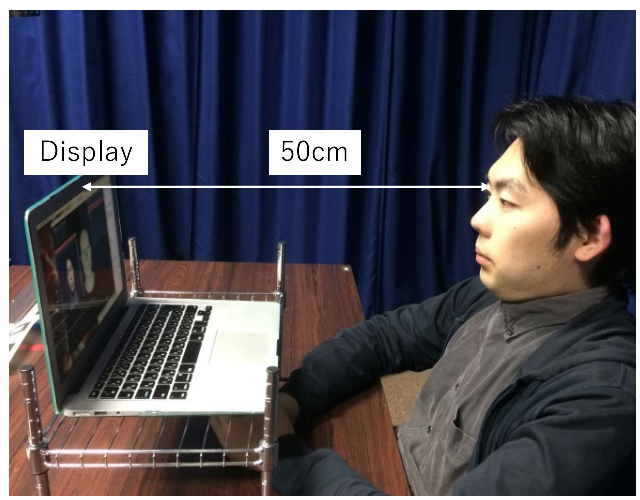


Figure 3. The experiment setup in this study.

II. Related Work

Measuring the degree of similarity between facial images is an important task in the field of face recognition. Many studies have used dimensional reduction techniques between feature representations to infer similarity. These studies include Eigenfaces [12] and Fisherfaces [13], which have been used in traditional face recognition.

An attempt to construct a facial similarity map was made by Holub et al. (2007). This map was calculated based on Triplets [14]. The resulting map demonstrated that the resulting map can effectively create a metric space that preserves notions of facial similarity.

The drawback of existing automatic facial expression recognition methods is that most of them focus on AU detection. Sadovnik et al. (2018) suggested that face recognition and face similarity are correlated, but the two are inherently different [15]. Since similarity in facial images can be recognized even when they are not the same person, there is a need to build a new dataset corresponding to the similarity of facial expressions. Vemulapalli and Agarwala (2019) built a large faces-in-the-wild dataset that labels facial expressions based on human visual preferences [16] and visualized the similarity of faces using t-SNE [17].

As described above, many studies have analyzed face similarity using 2-Dimensional (2D) facial images. However,

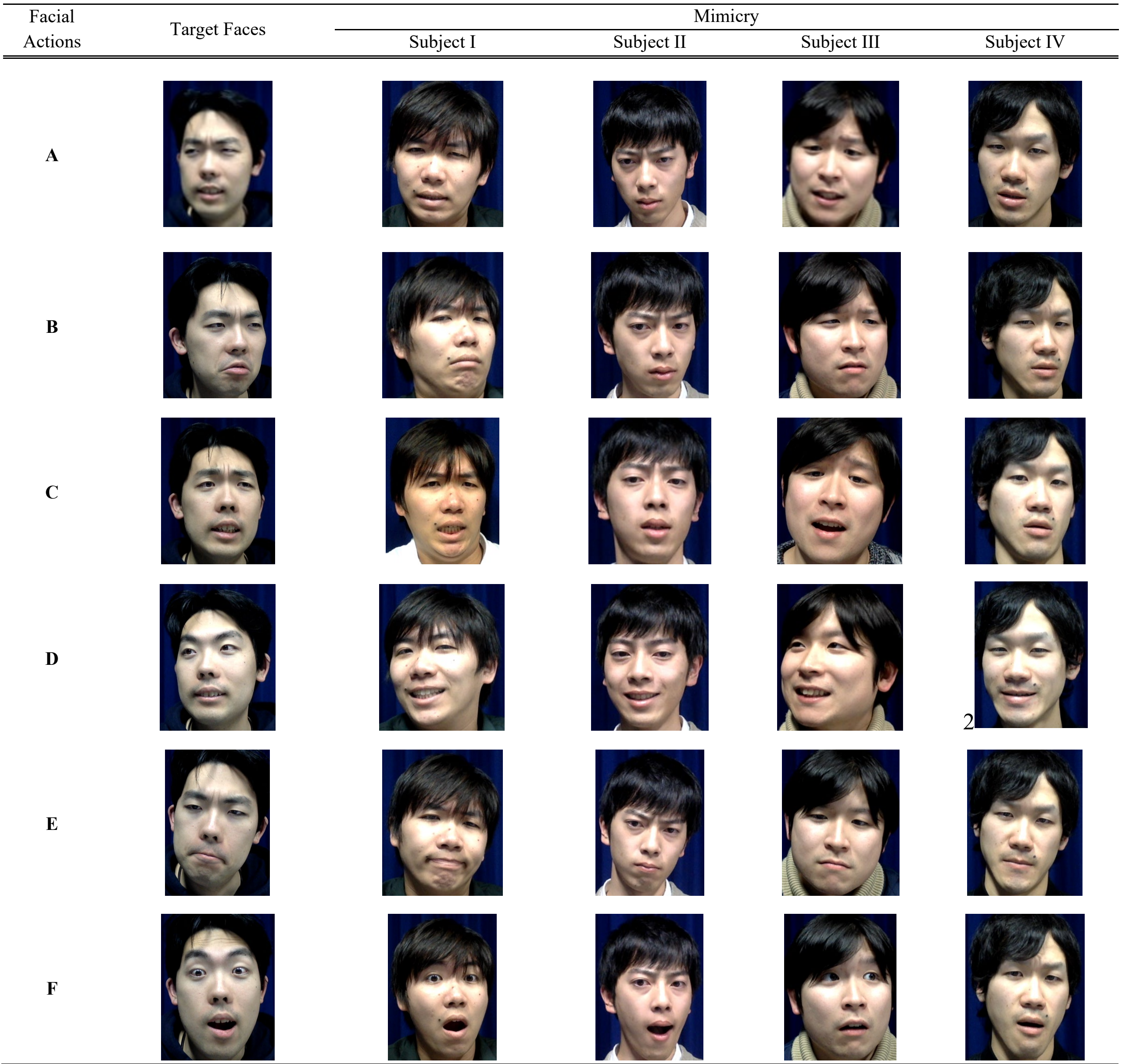


Figure 4. The target face and facial mimicry performed by each subject.

only few studies have used 3D facial images. Moorthy et al. (2010) extracted Gabor features from automatically detected fiducial points of texture images from the 3D face and demonstrate that these features correlate well with human judgements of similarity [18].

After Blanz and Vetter (1999) introduced a 3DMFM as a general face representation and a principled approach to image analysis [19], 3DMFM have incorporated into many solutions for facial analysis. 3DMFM generates a 3D face data from a given facial image and modifying the shape and texture in a natural way using its Principal Component Analysis (PCA) model of face shape and color information. Huber et al (2016) have made a publicly available 3DMFM, accompanied by their open-source software framework [11].

Another technique to generate an associate 3D face data from a given facial image is the deformation transfer. It is a well-recognized technique in computer graphics that creates expressive and plausible animations. Sumner and Popović (2004) proposed deformation transfer for triangle meshes, where the cloning process does not require the source and the target model to share several vertices or triangles [10]. Prima et al. (2020) demonstrated deformations of faces based on their expressions [1] as an initial attempt to measure similarity of pairs of 3D faces (Figure 1).

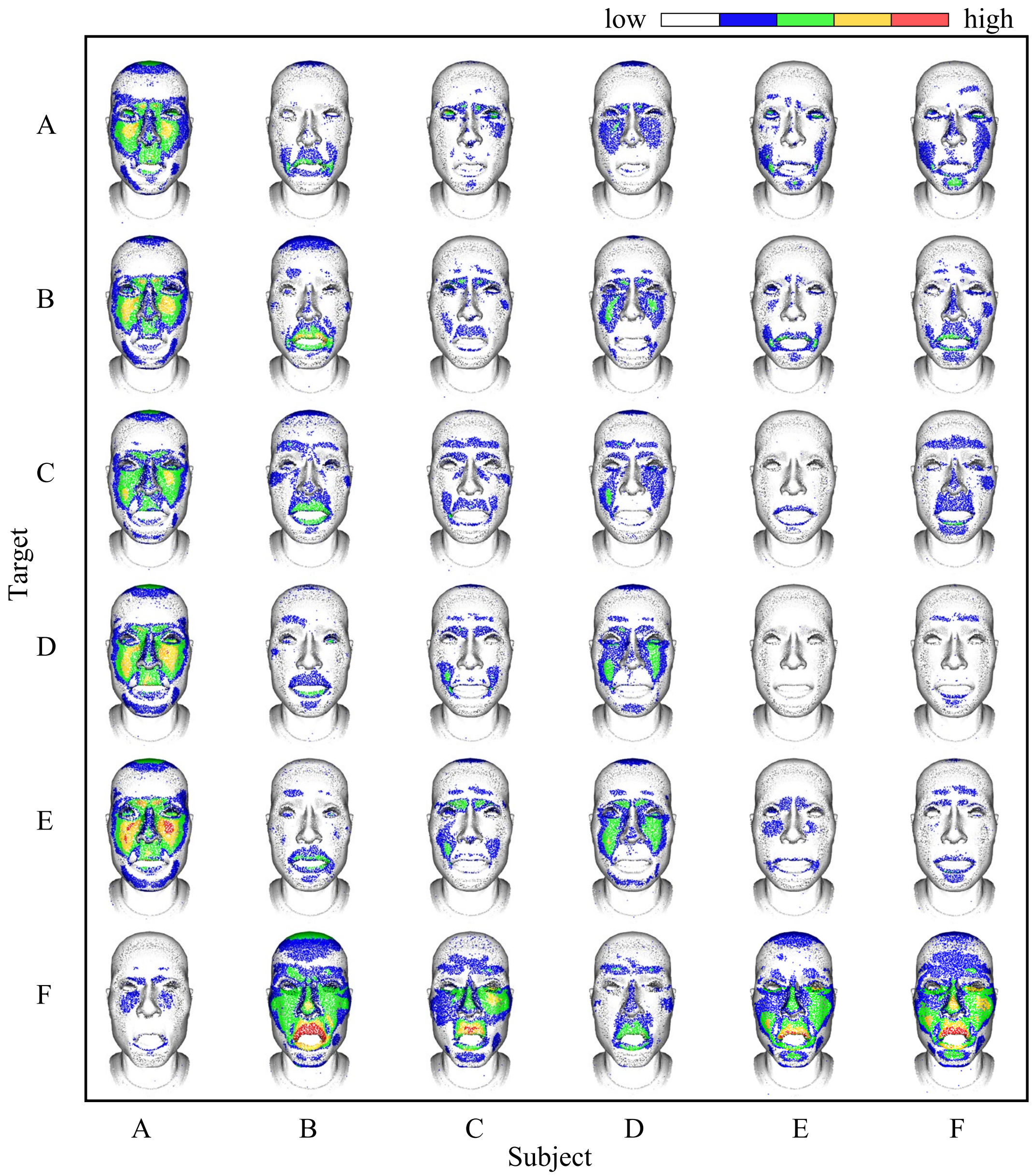


Figure 5. Differences in distance between the point clouds of 3DDC for the target face and the mimetic face of subject I.

To fit a 3D face data into a facial image, some points of the model need to be associated to the corresponding 2D landmark points in the facial image. Extracting landmark points from facial images can be done using automated facial landmarks tools, such as Dlib library [20]. Those 2D landmark points are mapped into the 3D face data using a shape-to-landmarks fitting method [21].

III. Facial Mimicry Analysis

Our attempt to analyze the facial mimicry of a pair of facial images takes three steps. The first step is to generate 3D face data from the facial images. The resulting 3D face data is then registered in a common 3D coordinate space. The second step is to sampling point clouds from the 3D face data. In the last step, we verify the similarity of the faces by analyzing the point clouds that correspond to each face. Each of these steps is described below.

1) *STEP 1: Generation of 3D Face Data*

Here, 3DDC and 3DMFM were generated from the input face images, respectively. Figure 2 shows the flow for generating the 3D face data in this study. At first, 68 facial landmark points were extracted from the face image using the Dlib library. The 3D face data was then fitted to these landmark points to generate 3D face data that represents the original face image. For the 3DDC, the shape-to-landmark

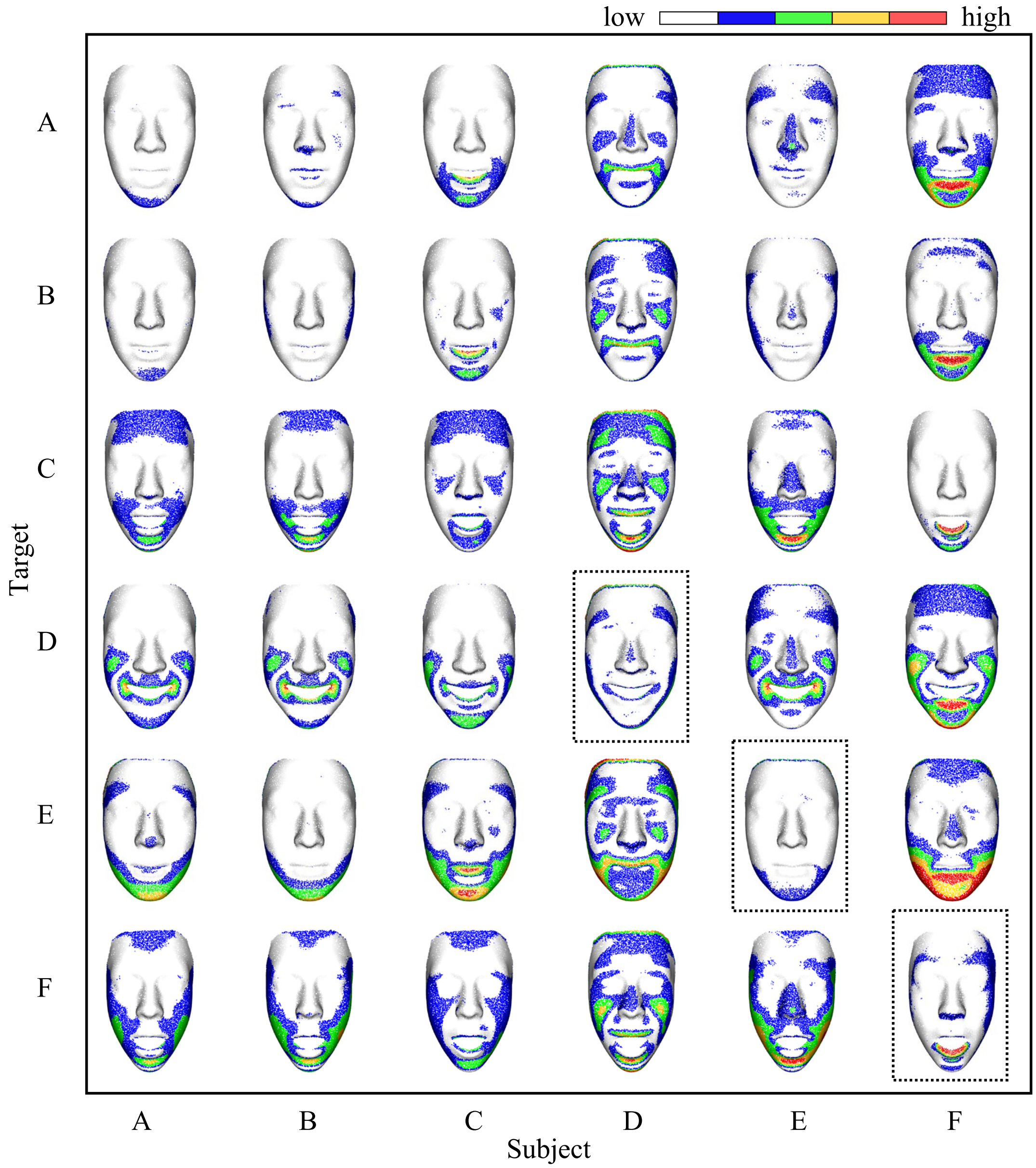


Figure 6. Differences in distance between the point clouds of 3DMFM for the target face and the mimetic face of subject I.

calculation was performed by solving the Perspective-n-Point (PnP) problem and deformation transfer [10]. The resulting rotation vector from the PnP were applied to the 3DDC so that the 3DDC was oriented the same as the input facial image. For the 3DMFM, pose estimation and shape fitting was performed using a lightweight 3DMFM fitting library, "eos" [22]. The generated 3D face data is saved in OBJ file format.

2) STEP 2: Point Cloud Extraction

We extracted point clouds from the resulting 3D face data obtained from the previous step. Open3D [23], an open-source library for analyzing 3D data, was used for this purpose. Sampling was done uniformly from the 3D surface based on the triangle area. Here, due to the different scales of 3DDC and 3DMFM, the sampling interval was adjusted to have the same number of sampling data from both face data.

3) STEP 3: Similarity Analysis

In order to analyze the similarity between facial images, the distance of each point on the original face was calculated to determine the distance to the target face. The internal functions of Open3D were used to calculate the distance. Here, the shorter the distance, the more similar the faces are.

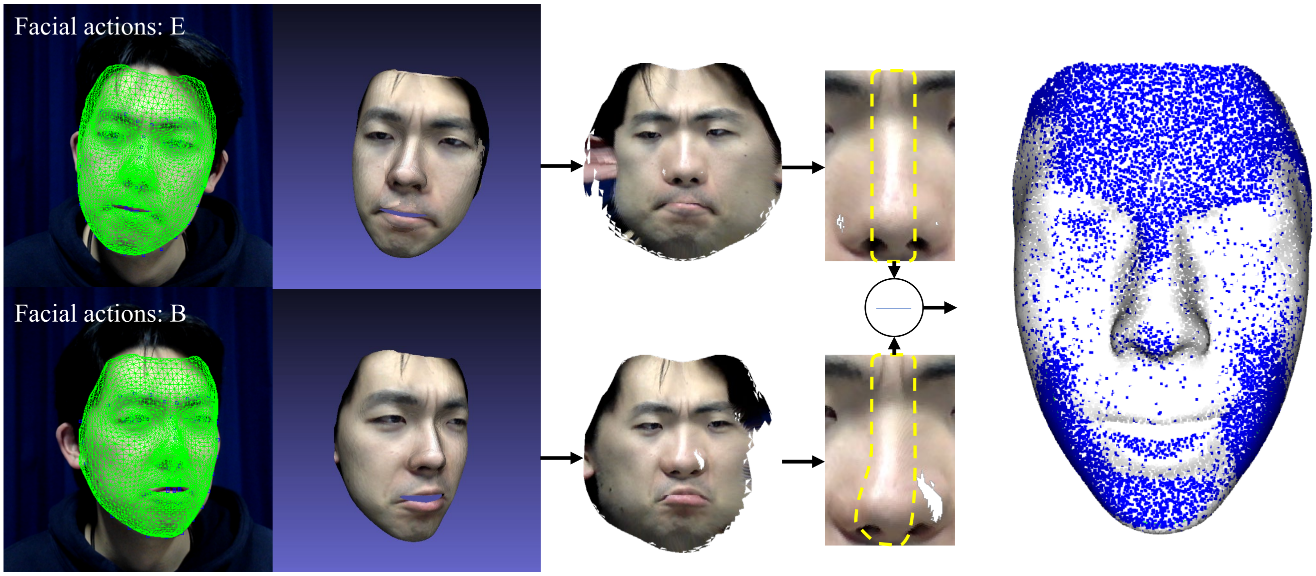


Figure 7. Distortions around the nose after the face frontalization.
(from left: 3DMFM aligned to the target face, face cropping, face frontalization (isomap), and distortion at the nose)

Considering that the non-rigid motion of the face is dominant when mimicking the opponent's face, we cropped the mouth and analyzed the correlation between the point clouds of this area.

IV. RESULT

Four male subjects (mean age 20.6 years) were recruited for the experiments. All subjects agreed to participate and signed the consent forms, to allow their data to be used in publications of this study. The room was set up with a table and stool for the subjects. On the table was a laptop computer that was used to display the target face to be mimicked by the subject. The monitor resolution was set at 1,440 × 900 and the refresh rate was 60 Hz. The subject was seated at approximately 50 cm from the laptop. Figure 3 shows the experimental setup in this study.

Subjects completed one practice block followed by six experimental blocks. For every block, a target face to mimic was displayed on the laptop. Subjects were asked to press the space button on the keyboard when they best imitate the target's face image. By pressing this button, the computer's built-in camera will take a picture of the subject's face. Finally, subjects were debriefed by the experimenter about the purpose of the study. Figure 4 shows the target face and facial mimicry performed by each subject. Subjects are imitating not only the facial expression of the target, but also the head posture of the target face.

1) Differences in distance between the 3D face data

3DDC and 3DMFM were used to generate 3D face data for target and mimetic faces. The difference in distance between the point clouds of 3D face data for the target face and the mimetic face of each subject was calculated. In order to be able to compare the two data correctly, we performed frontalization to each 3D face data in advance.

Figures 5 and 6 show the calculation results for 3DDC and 3DMFM of subject I, respectively. The colors indicate the degree of difference between the target face and the imitated face. Ideally, there will be the least difference between the two 3D face data in the right downward diagonal direction. In Figure 5, no facial actions yield the most similar (least different) results in that direction. However, in Figure 6, facial action: D, E, and F (dotted rectangles) were found to be the most similar between pairs, indicating subject I properly mimic the given three target faces.

In the case of the 3DDC face data, there are large differences in non-rigid areas such as the forehead, cheeks, eyebrows, and mouth. Similar results were seen in the 3DMFM face data, but the color distribution appeared to be less. Interestingly, there were significant differences at around the area of the nose, which is the rigid area. To figure out what caused these differences, we observed the 3D face data generated from the same person but in different head poses.

Figure 7 shows the 3D face data generated from the two target faces: B and E. As described in the previous section, we applied the initial 3DMFM to these faces using the shape-to-landmark method. However, after the 3D face data was frontalized, the difference in the shape of the nose are observable even in the same person's face. This suggests that the resulting frontalized face data contain some degree of distortion. Therefore, 3DMFM may not be supposed to generate 3D facial data with extreme head postures. This drawback also applies to 3DDC.

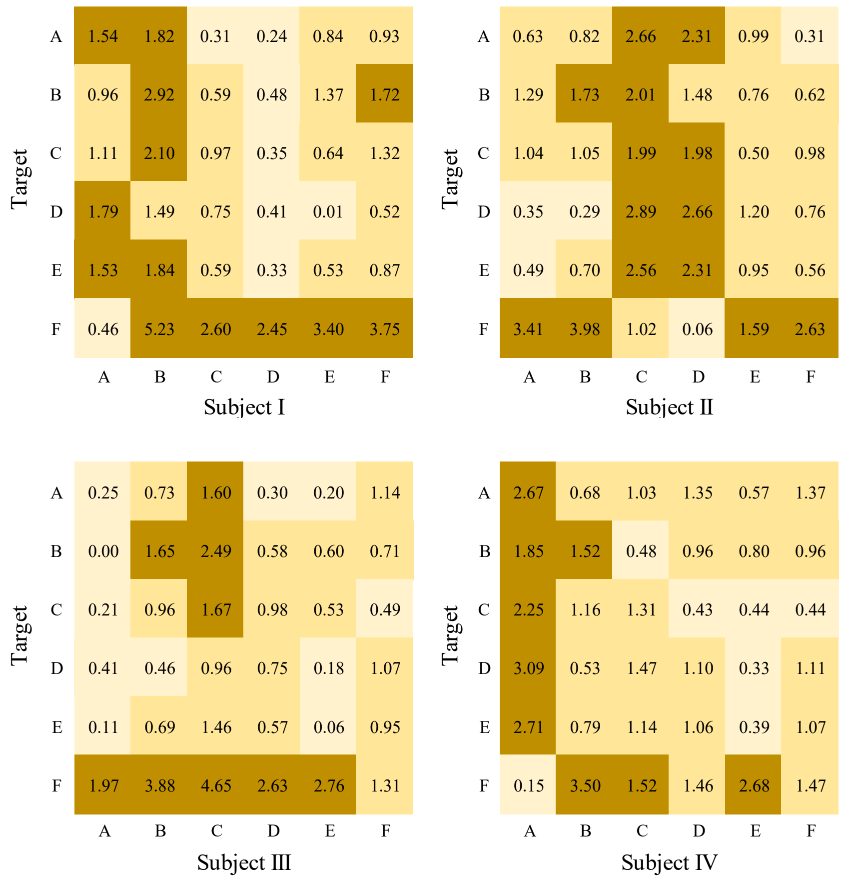


Figure 8. Differences in distance among 3DDC of the target faces and the mimetic faces of all subjects.

Figure 8 shows the difference in distance between the 3DDC of the subject faces and the 3DDC of the mimetic faces of all subjects. Similarly, Figure 9 shows the difference in distance between the 3DMFM of the subject faces and the 3DMFM of the mimetic faces of all subjects. Values were standardized to enable comparison among different data sources: 3DDC and 3DMFM. These values were colored in three levels to make them visually distinguishable. The values surrounded by dotted rectangles indicate the smallest difference between the presented face image and the imitated face image. To summarize, the 3DMFM sees that subjects I and III imitated the target faces of C, D and E, whereas, subject II imitated the target faces of B, C and E. Unfortunately, we were not able to see the suitability of facial mimicry in 3DDC just by calculating the difference in distance of the 3D face data. Overall, these results indicated that measuring the similarity between two 3D face data based solely on the differences between them is not sufficient.

2) Correlation Analysis between the 3D face data

For a more detailed analysis, we analyzed the similarity between two 3D facial data for point clouds in and around the mouth. Here, since the 3DDC is a complete head model, a 3D bounding box was needed to crop the mouth region. Open3D was used to perform this task.

Table I shows the correlations between the target face and the mimetic face in the mouth region of 3DDC for all subjects, whereas Table II shows the correlations in the mouth region of 3DMFM. In the 3DDC, the correlation was highest when the subject's facial expression was the same as that of the presenting face (values surrounded by a square on the diagonal). However, it might be difficult to assess facial mimicry solely based on this value because of the overall high values of the correlation coefficients in all data, as shown in Table I. On the other hand, the values of correlation coefficients obtained from the 3DMFM varied, but the correlation coefficients were found to be high when the target face's facial action were consistent with those of the subject. This finding indicates that 3DMFM is more suitable than 3DDC for evaluating facial mimicry.

3) Perceptual Judgment

In order validate the resulted facial mimicry performed by each subject, a 5-point Likert scale was used to gauge similarity. A value of 1 indicated that the two faces were completely dissimilar, while a value of 5 indicated that the two faces being compared were identical. Ten independent

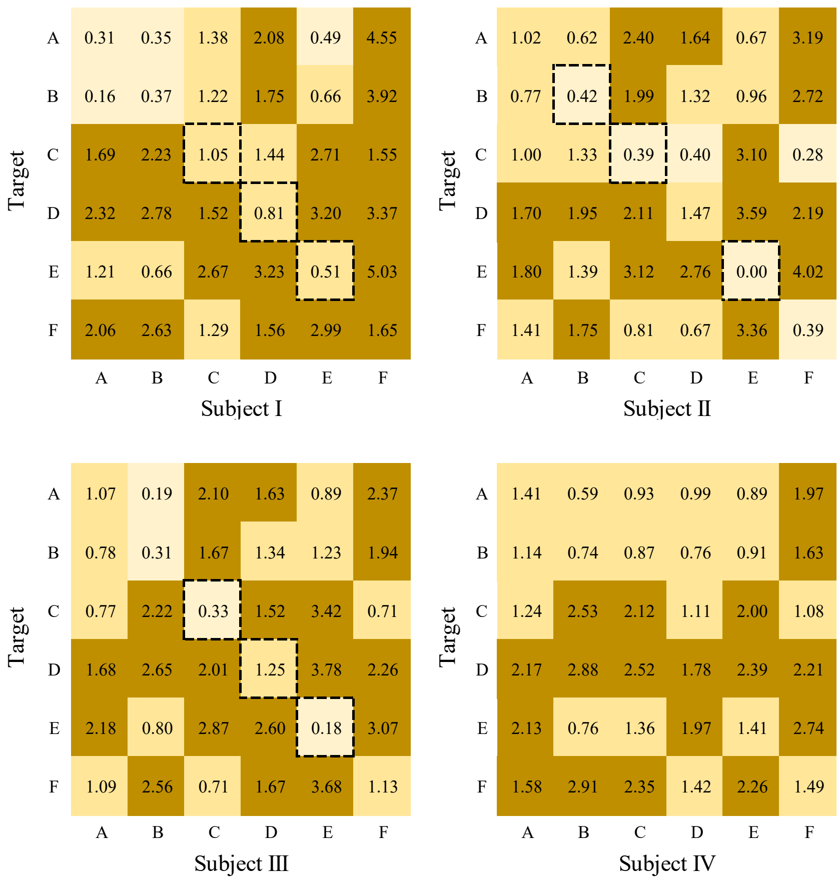


Figure 9. Differences in distance among 3DMFM of the target faces and the mimetic faces of all subjects.

raters were asked to rate the similarity between their faces and the presented faces shown in Figure 4. The raters were instructed to focus on the three areas of the eyebrows, eye shape, and mouth, which varied greatly.

To evaluate whether there was a difference in the mean score on the perceived similarity of pairs of faces, we conducted a one-way ANOVA. The means and standard deviations are shown in Table III. These results showed that there was a significant effect of the type of the facial action on the perceptual judgment level for the six conditions [$F(5, 234) = 2.25$, $p < 0.01$]. The target face and the face imitated by the subject were perceived to be similar, except for the facial action C.

V. Concluding Remarks

In this study, we extended our previous work to assess the similarity of pairs of 3D face data by using differences in the distance and correlation of point clouds in the 3D face data. The face data use in our study was generated using deformable 3-Dimensional Digital Character (3DDC) and the Surrey 3-Dimensional Morphable Face Model (3DMFM). We utilized the deformation transfer technique to clone the subject's facial movements into the 3DDC when mimicking a given facial image. The 3DMFM allows for a closer correspondence between the points, rather than only a set of facial feature points as in 3DDC.

To analyze the similarity of pairs of 3D face data, all 3D face data generated in this study were frontalized. We believed that comparison of faces can be made more effective by using a face-frontal view. However, while analyzing the experimental data, we noticed that some facial parts were distorted by this process. Therefore, we suggest that we need to be aware of this distortion when generating 3D facial data with extreme head posture.

Our experimental results show that measuring similarity using only the differences between point clouds of 3D face data is not sufficient. However, for the non-rigid part of the face, the similarity between the two 3D facial data can be measured by performing correlation analysis.

The proposed system can be extended as a face imitation training tool to improve social communication. Using this tool, users can practice imitating faces from photographs by referring to the differences in the reconstructed 3D facial data.

Acknowledgment

This work was supported by the grant of the Iwate Prefectural University's strategic project. We also thank the

TABLE I. CORRELATIONS BETWEEN THE TARGET FACE AND THE MIMETIC FACE IN THE MOUTH REGION OF 3DDC.

Subject I		A	B	C	D	E	F
Target	A	0.99	0.91	0.88	0.9	0.87	0.92
	B	0.91	0.99	0.89	0.92	0.89	0.93
	C	0.89	0.91	0.99	0.93	0.92	0.93
	D	0.91	0.93	0.93	0.99	0.92	0.93
	E	0.87	0.9	0.92	0.93	0.99	0.92
	F	0.89	0.93	0.91	0.91	0.9	0.99

Subject II		A	B	C	D	E	F
Target	A	0.99	0.91	0.88	0.89	0.85	0.91
	B	0.91	0.99	0.89	0.91	0.87	0.93
	C	0.89	0.91	0.98	0.92	0.91	0.93
	D	0.91	0.92	0.93	0.99	0.91	0.92
	E	0.87	0.9	0.93	0.93	0.99	0.91
	F	0.89	0.93	0.91	0.9	0.88	0.99

Subject III		A	B	C	D	E	F
Target	A	0.99	0.92	0.89	0.91	0.87	0.91
	B	0.91	0.99	0.90	0.92	0.89	0.93
	C	0.90	0.91	0.99	0.93	0.92	0.93
	D	0.92	0.93	0.93	0.99	0.92	0.92
	E	0.88	0.9	0.93	0.93	0.99	0.91
	F	0.9	0.93	0.92	0.91	0.9	0.99

Subject IV		A	B	C	D	E	F
Target	A	0.99	0.91	0.9	0.9	0.85	0.91
	B	0.91	0.99	0.90	0.91	0.88	0.93
	C	0.89	0.90	0.99	0.93	0.91	0.93
	D	0.91	0.92	0.94	0.99	0.91	0.92
	E	0.87	0.90	0.93	0.93	0.99	0.91
	F	0.89	0.92	0.92	0.90	0.89	0.99

TABLE II. CORRELATIONS BETWEEN THE TARGET FACE AND THE MIMETIC FACE IN THE MOUTH REGION OF 3DMFM.

Subject I		A	B	C	D	E	F
Target	A	0.99	0.67	0.71	0.68	0.6	0.66
	B	0.67	0.99	0.68	0.76	0.74	0.59
	C	0.74	0.66	0.99	0.69	0.6	0.77
	D	0.68	0.73	0.71	0.99	0.69	0.62
	E	0.61	0.76	0.61	0.73	0.99	0.44
	F	0.76	0.68	0.79	0.68	0.56	0.97

Subject II		A	B	C	D	E	F
Target	A	0.99	0.68	0.73	0.69	0.64	0.72
	B	0.67	0.99	0.70	0.77	0.76	0.66
	C	0.75	0.68	0.99	0.69	0.64	0.78
	D	0.68	0.74	0.73	0.99	0.72	0.68
	E	0.60	0.75	0.65	0.75	0.99	0.53
	F	0.77	0.70	0.78	0.68	0.6	0.99

Subject III		A	B	C	D	E	F
Target	A	0.98	0.69	0.71	0.7	0.65	0.64
	B	0.66	0.99	0.68	0.74	0.77	0.56
	C	0.75	0.68	0.99	0.73	0.64	0.75
	D	0.68	0.74	0.71	0.99	0.72	0.6
	E	0.58	0.75	0.62	0.70	0.99	0.41
	F	0.77	0.71	0.78	0.72	0.6	0.96

Subject IV		A	B	C	D	E	F
Target	A	0.93	0.72	0.73	0.72	0.67	0.72
	B	0.61	0.98	0.69	0.76	0.76	0.66
	C	0.76	0.73	0.99	0.75	0.72	0.79
	D	0.64	0.75	0.72	0.99	0.74	0.68
	E	0.51	0.72	0.63	0.72	0.94	0.53
	F	0.79	0.76	0.79	0.74	0.67	0.99

editor and three anonymous reviewers for their constructive comments, which helped us to improve the manuscript.

TABLE III. THE MEANS AND STANDARD DEVIATIONS OF THE PERCEPTUAL JUDGEMENT.

	FACIAL ACTION					
	A	B	C	D	E	F
MEAN (STDEV)	2.9 (1.16)	3.3 (1.10)	2.2 (1.19)	4.2 (0.94)	4.0 (0.88)	3.8 (1.07)

REFERENCES

[1] O. D. A. Prima, H. Ito, T. Tomizawa, and T. Imabuchi, “Facial Mimicry Training Based on 3D Morphable Face Models,” The Thirteenth International Conference on Advances in Computer-Human Interactions, ACHI 2020, pp. 57-60, 2020.

[2] J. L. Lakin, V. E. Jefferis, C. M. Cheng, and T. L. Chartrand, “The Chameleon Effect as Social Glue: Evidence fot the Evolutionary Significance of Nonconscious Mimicry,” Journal of Nonverbal Behavior, 27(3), pp. 145–162, 2003.

[3] P. Ekman et al., “Universals and Cultural Differences in the Judgments of Facial Expressions of Emotion,” Journal of Personality and Social Psychology, 53(4), pp. 712-717, 1987.

[4] M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Measuring Facial Expressions by Computer Image Analysis,” Psychophysiology, Cambridge University Press, 36(2), pp. 253–263, 1999.

[5] Y. L. Tian, T. Kanade, and J. F. Cohn, “Recognizing Lower Face Action Units for Facial Expression Analysis.” Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000, 23(2), pp. 484–490, 2000.

[6] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L-P. Morency, “OpenFace 2.0: Facial Behavior Analysis Toolkit,” IEEE International Conference on Automatic Face and Gesture Recognition, 2018.

[7] iMotions, https://imotions.com/ [retrieved: August 31, 2020]

[8] G. Littlewort et al., "The computer expression recognition toolbox (CERT)," Face and Gesture 2011, Santa Barbara, CA, 2011, pp. 298-305, doi: 10.1109/FG.2011.5771414.

[9] D. McDuff et al., “AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit,”

Conference on Human Factors in Computing Systems, pp. 3723–3726, 2016. https://doi.org/10.1145/2851581.2890247

[10] R. W. Sumner and J. Popović, "Deformation Transfer for Triangle Meshes," ACM Transactions on Graphics, 23(3), pp. 399-405, 2004.

[11] P. Huber et al., "A Multiresolution 3D Morphable Face Model and Fitting Framework," 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, pp. 79–86, 2016.

[12] M. A. Turk and A. P. Pentland. Face Recognition Using Eigenfaces. In Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on, pp. 586–591. IEEE, 1991.

[13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigen faces vs. Fisherfaces: Recognition using class specific linear projection," IEEE Transactions on pattern analysis and machine intelligence, 19(7), pp. 711–720, 1997.

[14] A. Holub, Y. Liu, and P. Perona, "On Constructing Facial Similarity Maps," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 17–22, 2007.

[15] A. Sadovnik, W. Gharbi, T. Vu, and A. Gallagher, "Finding Your Lookalike: Measuring face similarity rather than face identity," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 2235–2353, 2018.

[16] R. Vemulapalli and A. Agarwala, "A Compact Embedding for Facial Expression Similarity," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 5683–5692, 2019.

[17] V. D. M. Laurens and G. Hinton, "Visualizing Data Using t-SNE" Journal of Machine Learning Research, 9, pp. 2579–2605, 2008.

[18] A. K. Moorthy, A. Mittal, S. Jahanbin, K. Grauman and A. C. Bovik, "3D Facial Similarity: Automatic Assessment Versus Perceptual Judgments," 2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington, DC, pp. 1-7, 2010.

[19] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," In ACM Transactions on Graphics (Proceedings of SIGGRAPH), pp. 187–194, 1999.

[20] V. Kazemi and J. Sullivan, "One Millisecond Face Alignment with an Ensemble of Regression Trees," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867–1874, 2014.

[21] O. Aldrian and W. A. P. Smith, "Inverse Rendering of Faces with a 3D Morphable Model," IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(5), pp. 1080–1093, 2013.

[22] eos: A Lightweight Header-Only 3D Morphable Face Model fitting library in modern C++11/14.
https://github.com/patrikhuber/eos
[retrieved: August 31, 2020]

[23] Open3D, http://www.open3d.org/
[retrieved: August 31, 2020]

A Pointing Device for 3D Interactive Spherical Displays

Oky Dicky Ardiansyah Prima, Katsuyoshi Hotta, Rintaro Takahashi, Hisayoshi Ito
Graduate School of Software and Information Science, Iwate Prefectural University
152-52 Sugo Takizawa, Japan
email: prima@iwate-pu.ac.jp { g236q004, g231r019 }@s.iwate-pu.ac.jp, hito@iwate-pu.ac.jp

***Abstract*—Three-Dimensional (3D) displays have been developed to allow users to view 3D objects from any angle. There are several input interfaces for interacting with this display, such as joysticks and gesture interfaces. However, in order to properly interact with the 3D objects in the display, the input interface needs to obtain its own 3D location from the 3D tracking device. In this study, we propose a low-cost pointing device for a 3D interactive display using an infrared camera and a smartphone instead of a 3D tracking device. The camera is set up at the bottom of the display to detect the contact point between the pointing device and the display surface. The pointing device gets its 3D orientation from the smartphone's built-in Inertial Measurement Unit (IMU). The proposed pointing device enables the user to see the auxiliary line from the device tip from different angles. We demonstrated some 3D pointing applications, such as selecting objects inside the display to show the usability of the proposed pointing device.**

Keywords-VR; 3D stylus; spherical display; virtual reality; perception.

I. INTRODUCTION

The increasing use of Three-Dimensional (3D) contents in the media and entertainment industry is providing a catalyst for the development of devices to effectively represent and interact with this content. This study extends our previous research on a 3D pointing device [1]. Over the past decade, this new development has boosted and refreshed interest in Virtual Reality (VR). VR devices such as the Oculus Rift and HTC Vive [2], which are capable of full-body tracking, provide users with a natural interaction with 3D contents in a virtual space.

Apart from VR devices, the development on 3D displays has taken a big step forward with the introduction of light field displays. These displays are generally intended to create motion parallax and stereoscopic disparity, allowing the observer to perceive the scene in 3D without the need to wear obtrusive glasses [3]. Looking Glass [4] is a currently available holographic display system that generates 45 distinct views of a 3D scene using the light field technique. In the future, rapid advances in electronics, optics and photonics are bringing more true 3D display technologies to the market [5].

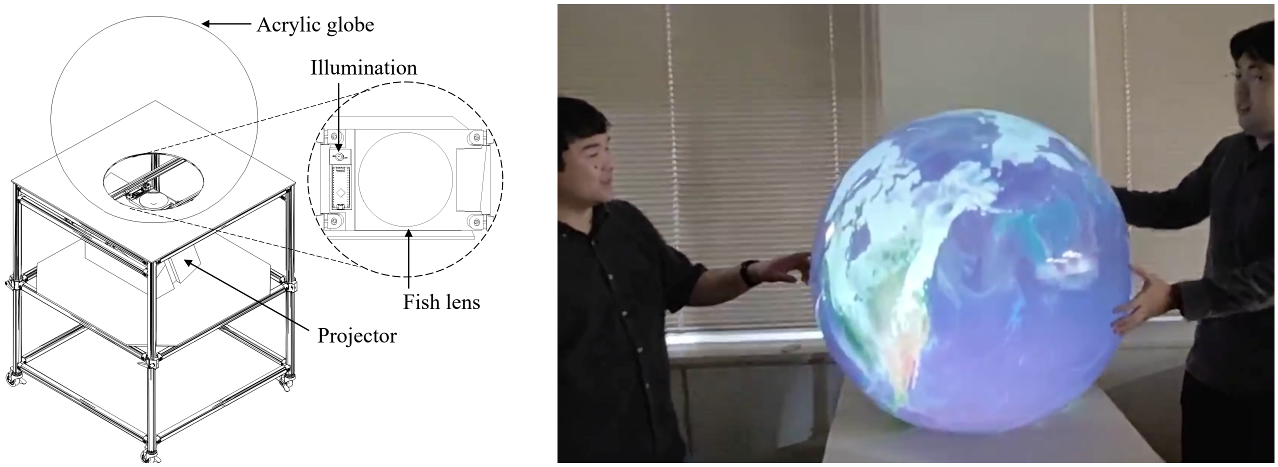
Another attempt to achieve true 3D representation is the use of non-planar displays instead of flat displays. 3D Spherical Display (3DSD) has an advantage on representing 3D objects to be seen from any angle. SnowGlobe [6] and Spheree [7] were developed as perspective-corrected 3DSDs based on a non-planar Two- to Three-Dimensional (2D-to-3D) mapping technique. SnowGlobe uses a single projector and a hemispherical mirror to reflect the image onto the display surface. In contrast, Spheree uses multiple calibrated projectors to project the image directly onto the display surface. Both displays use a 3D tracking device to track the location of the user's head relative to the display, providing a motion-parallax-corrected perspective of the objects on the display. CoGlobe [8][9][10], an upgrade of the Spheree, supports two users to view 3D contents from each own viewpoint using modified active shutter glasses. 3DSDs may have overcome the limited situation awareness of headset VR.

3DSDs can be equipped with a touch interface by optically tracking the user's rear illuminated fingers that appear as blob images when touching the display surface [11]. Each center of the blob has a 3D coordinate (*x*, *y*, *z*) originating from the center of the 3DSD. CoGlobe demonstrated several 3D games using 3D pointing devices such as pointing devices and rackets. The OptiTrack system [12] tracked the 3D location and orientation (*x*, *y*, *z*, *pitch*, *yaw*, *roll*) of multiple passive reflective markers attached to those devices. Hereinafter, the information about the 3D location and orientation is simply referred to as Six-Degree of Freedom (6DoF). This system also uniquely identified and labeled each group of markers attached to each device.

Attaching multiple reflective markers to a pointing device can degrade its functionality. In addition, in order to estimate the 6DoF, those markers needed to be connected to each other like branches, which increased the shape of the pointing device and made it difficult to use. Instead of reflective markers, Augmented Reality (AR) markers can be used to estimate the 6DoF of a pointing device, such as the DodecaPen [13]. However, due to the shape of the 3DSD, the pointing device is often hidden from the camera used to track the AR markers, making it impossible to estimate the 6DoF.

In this study, we propose a pointing device suitable for a spherical display by using a touch interface to acquire the 3D location and an Inertial Measurement Unit (IMU) to acquire the orientation. The orientation data is calibrated using the Motion Platform and the corrected data is sent to the computer controlling the 3DSD and combined with the 3D location obtained from the touch interface.

The rest of this paper is organized as follows. In section II, we describe the 3DSD hardware and software developed for this study. Section III introduces our approach to implement the pointing device. Section IV describes our experiment results. Section V discusses about the further enhancements to the proposed pointing device. Finally, Section VI presents our conclusions and future works.



(a) The hardware design of our 3DSD (b) The globe

Figure 1. Our 3D spherical display (a) and the globe (b) projected onto the display.

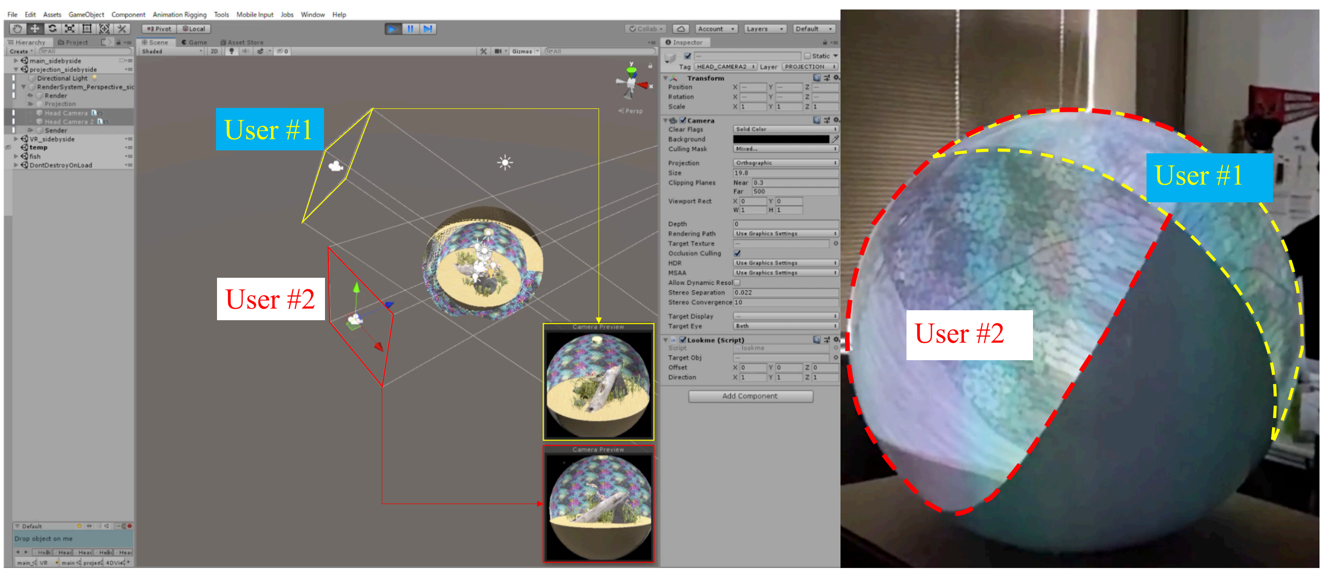


Figure 2. Our 3DSD software program based on the Unity 3D platform.

II. THE 3D SPHERICAL DISPLAY FOR THIS STUDY

A 3DSD was created for this study using a single projector. Figure 1(a) shows the hardware design of the 3DSD. We used a spherical acrylic lighting case (acrylic globe) as the display material. The interior of the display was painted with rear-projection acrylic paint, so that images projected from inside are visible from the outside of the display. A 3D projector with 4K resolution to was used to generate a high-resolution image onto the display. A fisheye lens was installed between the projector and the acrylic globe to spread the image from the projector to the entire surface of the interior of the spherical acrylic. In order to make the display touchable, we placed a wide-lens infrared camera at the bottom. Since the axes of the camera and of the fisheye lens are different, we aligned these axes using an affine transformation. For the infrared illumination, we installed a high-power infrared LED at the opposite side of the camera. Figure 1(b) shows the globe on our 3DSD.

To make it easier for users to visualize 3D objects, we have created a software program based on the Unity 3D Platform that allows any 3D application running on the platform to be displayed on the 3DSD. Figure 2 shows how two users view a 3D scene from different perspectives. As shown in the figure, the software program generated images for each viewpoint and projected those images onto the surface of the 3DSD. The two images appear to be superimposed, but each image is displayed at a different timing. Therefore, if the user uses shutter glasses, the images at each timing can be viewed separately. Currently, the HTC

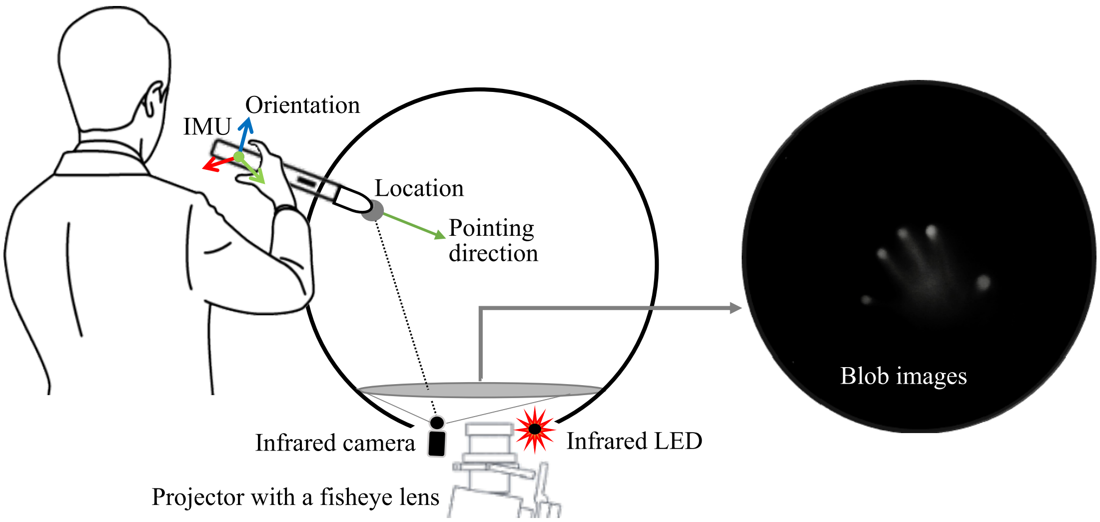


Figure 3. The mechanism of the proposed pointing device.

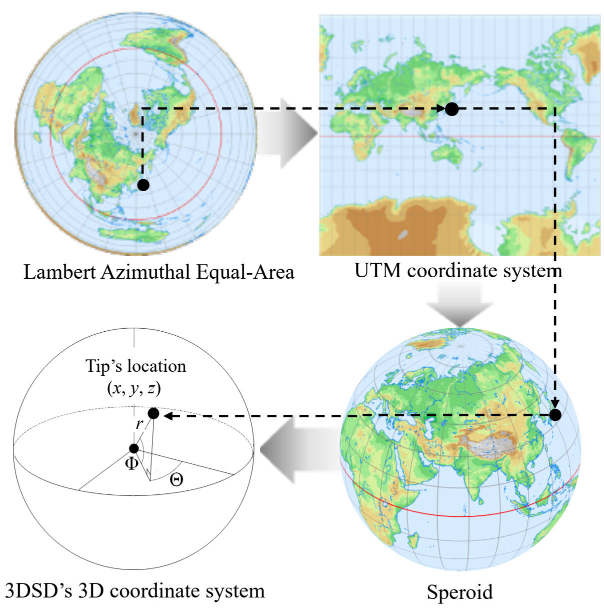


Figure 4. 3D coordinates of the tip of the pointing device.

Vive Tracker [14] needs to be attached to the user's head to determine the user's perspective. In the future, we plan to simplify the tracking system by replacing it with a 3D face tracking based on a vision camera.

III. Our Proposed Pointing Device

Figure 3 shows a scene in which the proposed pointing device is used on a 3DSD. The camera for the 3DSD touch interface captures any blob images on the 3DSD surface. The proposed pointing device does not use a 3D tracking device such as motion capture, but acquires the location based on the blob image detected by the infrared camera and the orientation based on the IMU. The details of the calculation to determine the location and orientation of the pointing device are as follows.

A. *Detecting the location of the tip of the pointing device*

When the tip of the pointing device touches the surface of the 3DSD, the blob image corresponding to the touched area is extracted to localize the tip's location. This process involves binarizing the blob image to form a contour at the boundary and fitting an ellipse to the contour. OpenCV (opencv.org), an open-source computer vision software library, is used for this calculation. Here, the coordinate of the tip of the pointing device is calculated as the center coordinate of the ellipse.

We perform the following coordinate transformations to convert the coordinates of the tip of the pointing device mentioned above into the 3DSD's 3D coordinate system. At first, since the image in the 3DSD is projected according to Lambert Azimuthal Equal-Area Projection, the coordinates of the tip of the pointing device are converted to this projection. Next, these coordinates are re-projected into the Universal Transverse Mercator (UTM) coordinate system and mapped onto spheroid, which represents the 3D coordinate system of the 3DSD. Figure 4 shows a series of the above transformations. The world map was used as the background image projected onto the 3DSD to make a better understanding of these transformations.

In our system, the blob image is extracted for everything that touches the surface of the 3DSD, but currently, we do not distinguish between the user's fingertip or the pointing device. Therefore, when using the prototype of this pointing device, the 3DSD should not be touched except by this device.

B. *Getting the rotation of the pointing device*

In this study, we prototyped the pointing device using a mobile device, the iPhone 7. As shown in Figure 5, we set up a pointing stick on the top of the iPhone 7 and attached a rubber to the tip to touch the 3DSD. The pointing device gets

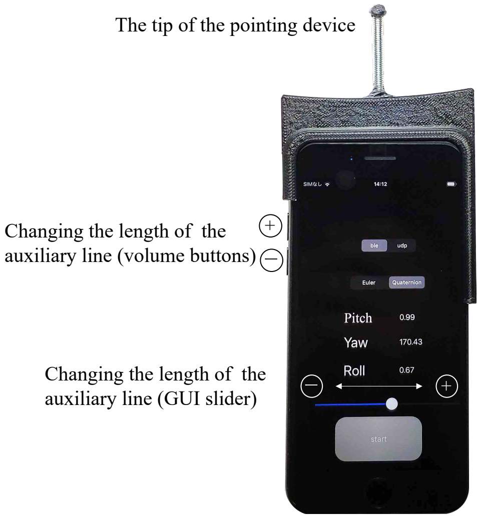


Figure 5. The proposed pointing device for the 3DSD.

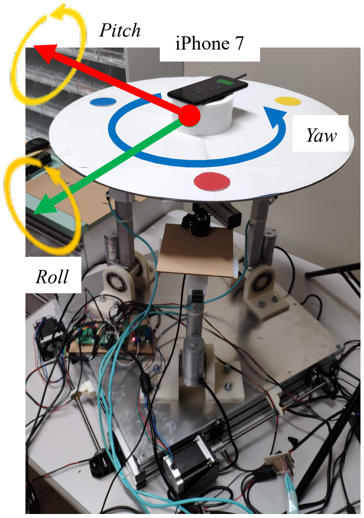


Figure 6. The Motion Platform built for this study.

its own orientation (*pitch*, *yaw*, *roll*) from the iPhone 7's built-in IMU. This information is transmitted to the 3DSD via Bluetooth of the iPhone 7. In this way, this integration of location and orientation information allows the pointing device to have 6DoF information. Using this information, the 3DSD software system will be able to calculate the posture of the pointing device and project an auxiliary line from its tip according to the user's perspective. Here, the length of the auxiliary line can be adjusted with the audio buttons on the iPhone 7 or the Graphical User Interface (GUI) on its screen.

The iPhone 7's built-in IMU includes a compass and a gyroscope sensor. However, to make the pointing device works properly, these sensor values must be calibrated in advance. In this study, we constructed a simplified Stewart Platform (Motion Platform) to calibrate the rotation information obtained from the IMU of the iPhone 7. Although this simplified platform has only three actuators instead of the six used in the original form, it is enough as a validation device for this study.

Figure 6 shows the calibration of the iPhone 7 on the Motion Platform. The calibration analyzes the difference between the rotation angle from the IMU and the rotation angle of the Platform. For calibration, the Motion platform was used to perform 27 arbitrary motions only on pitch and roll angles. The reason for not changing direction to the yaw angle is to measure the stability of the IMU when measuring the yaw angle. Since the changes in the yaw angle are the result of the changes in pitch and roll, there will be less changes in the yaw angle. Figure 7 shows orientation angles derived from the IMU after being calibrated using the Motion Platform. Figure 8 shows the relationship between the rotation angle of the Motion Platform and the rotation angles of the iPhone 7. Despite the variability of the pitch and roll angles of about ± 10 degrees, the variability of the yaw angle is limited

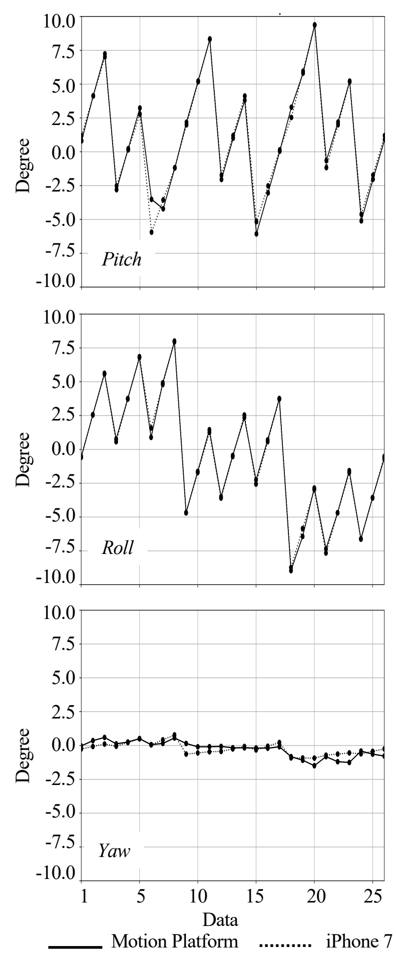


Figure 7. Values of orientation angles derived from the IMU after being calibrated using the Motion Platform.

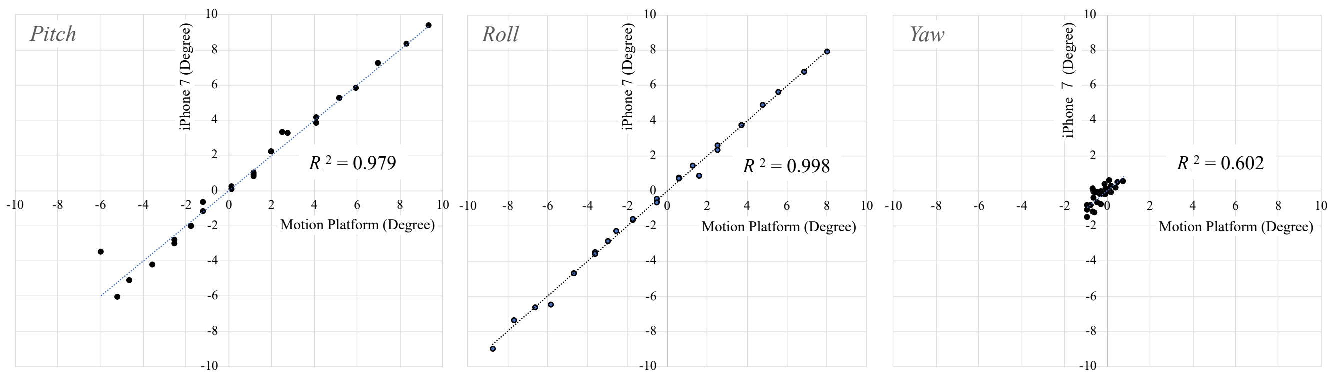


Figure 8. The relationships between orientation angles of the Motion Platform and the iPhone 7.

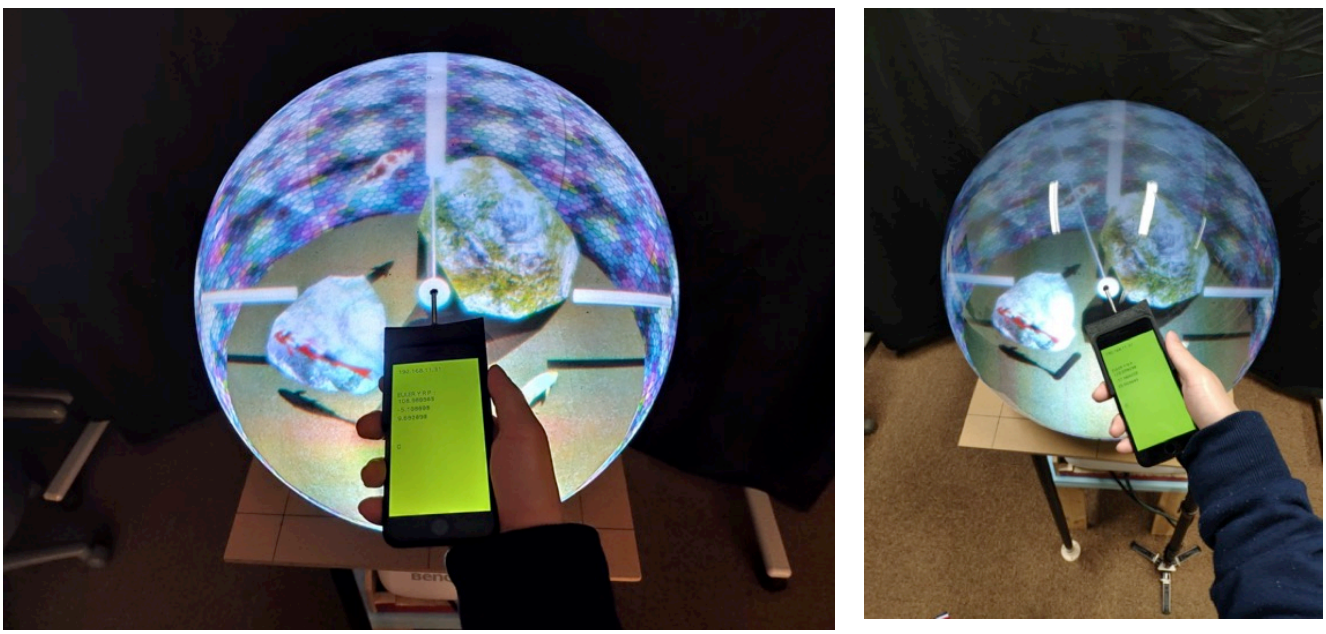


Figure 9. Our working pointing device.

to about ±2 degrees. There was a high correlation between the pitch angles, $r(25) = .99$, $p < .001$, with a $R^2 = .979$. Similarly, there was also a high correlation between the roll angles, $r(25) = .99$, $p < .001$, with a $R^2 = .998$, and between the yaw angles, $r(25) = .78$, $p < .001$, with a $R^2 = .602$. These results indicate that the IMU built into the iPhone 7 is accurate enough to perform interactions that require absolute orientation.

Figure 9 shows the proposed pointing device that works with the 3DSD. The HTC Vive Tracker was attached to the camera to capture this image from the correct perspective. To give users a strong impression of motion parallax, we attached a wallpaper pattern on the back of the virtual fish-tank. From the camera's perspective, the auxiliary line of the pointing device extends in the correct direction according to the orientation of the device. The volume button or slider as shown in Figure 5 allows the user to adjust the length of this line.

IV. Experimental Results

We evaluated the proposed pointing device in terms of accuracy, pointing stability, and user experience. A 51 cm diameter 3DSD was used in the experiment. The display, pointing device and user perspective coordinate systems were calibrated using the HTC Vive tracker. The 3DSD runs on a desktop computer with a 3.6 GHz CPU, 32 GB of RAM and a GTX980Ti graphics card.

A. Pointing Accuracy

In the 3DSD, the accuracy of the pointing device is required for operations such as selecting a 3D object. In the case of the 3DSD with a diameter of 51 cm, one degree of

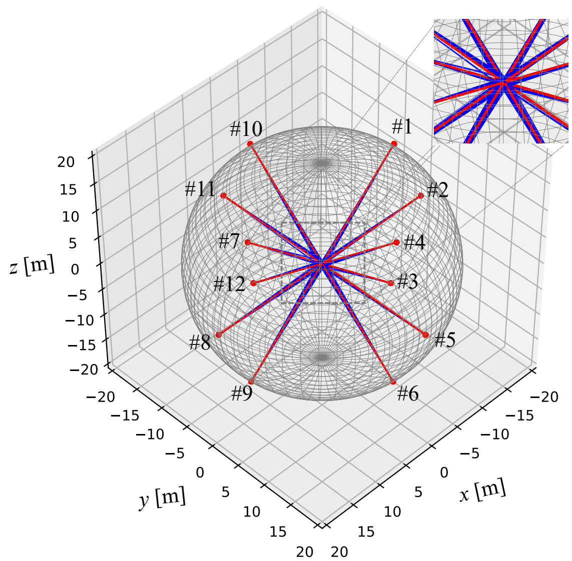


Figure 10. The resulting vectors (blue) from the pointing device tip and their corresponding ground truth (red).

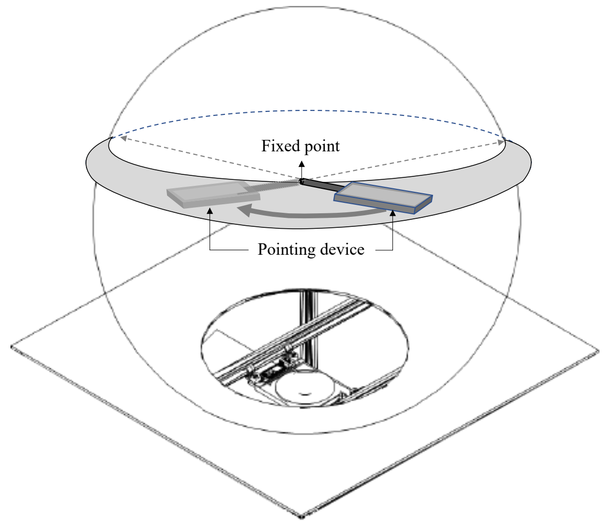


Figure 12. Schematic diagram of the pointing experiment in yaw angle direction.

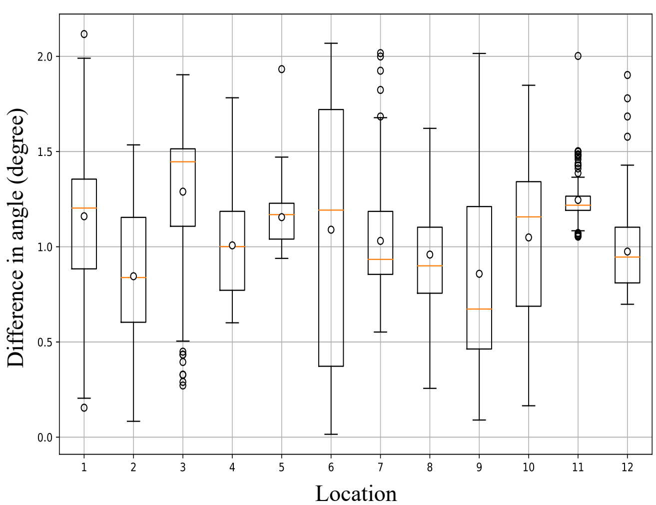


Figure 11. The distribution of the angle differences at each of the 12 locations.

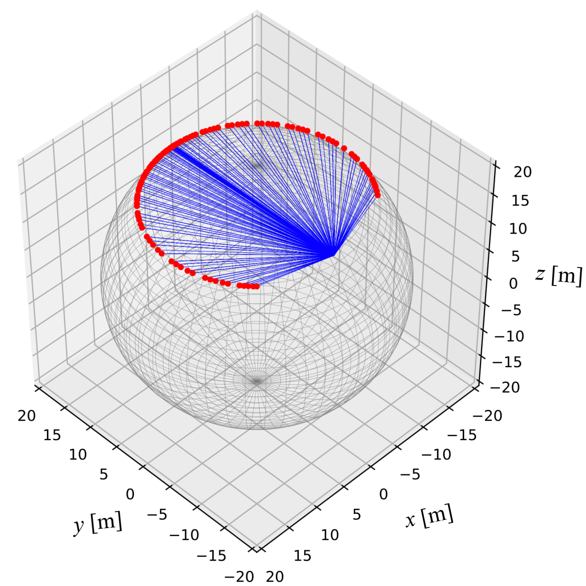


Figure 13. Intersection points of the vectors from the pointing device tip and the display surface.

pointing error corresponds to a deviation of about 1 cm at the center of the 3DSD. In order to evaluate the accuracy of pointing, 12 arbitrary locations on the 3DSD surface were selected, and the pointing device was set from these locations towards the center of the display. At each point, vectors representing the pointing direction were recorded for two seconds, and the difference with the vector connecting the pointing point and the center of the 3DSD, hereafter referred to as the ground truth vector, was analyzed. Each measurement was sampled at 60 Hz, resulting in 120 vectors.

Figure 10 shows the 12 locations used for the evaluation and vectors of the pointing device pointing to the center of the display from those locations (blue lines). The red lines show the ground truth vectors. A small discrepancy can be observed between the red and blue lines. Here, we took the inner product of both vectors and quantified the difference in angles. Figure 11 shows the distribution of the angle differences at

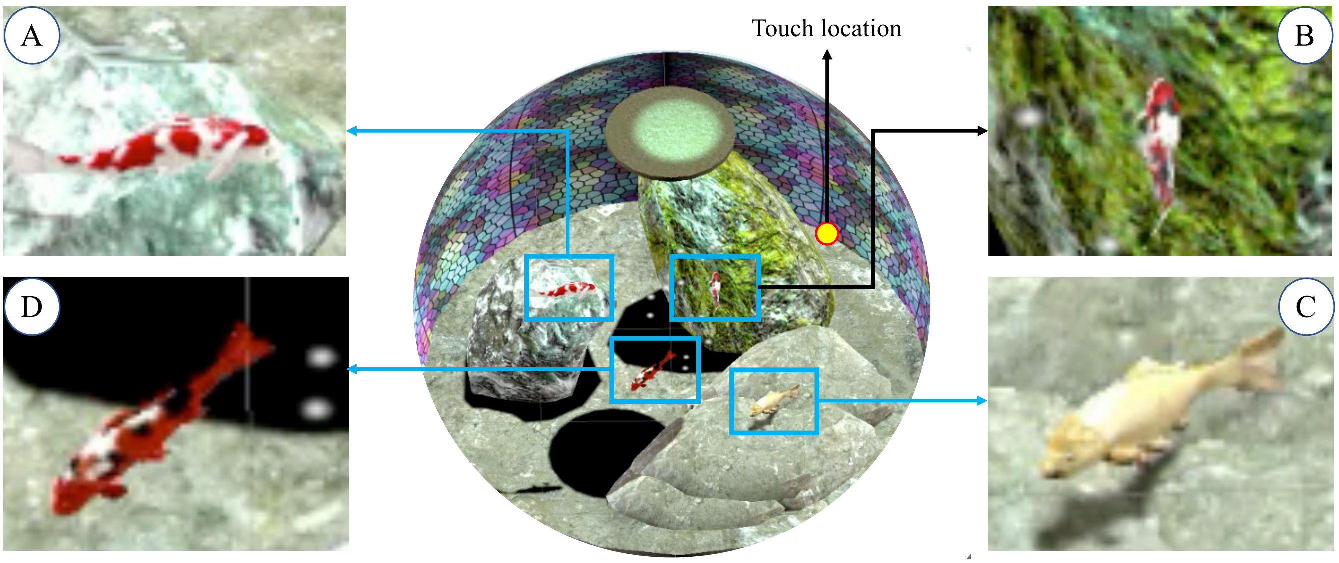


Figure 14. The virtual fish-tank for the experiment.

TABLE I. TIME BEING TAKEN BY THE SUBJECT TO POINT EACH FISH.

Subject ID	Fish A	Fish B	Fish C	Fish D
1	8.4	3.4	5.5	4.9
2	7.5	2.6	3.6	3.3
3	4.3	2.5	5.1	4.6
4	3.5	10.6	4.4	4.8
5	8.4	4.3	5.3	4.9
6	5.9	4.6	4.4	5.2
7	7.8	3.5	4.1	3.8
8	6.9	3.2	2.9	3.8
9	10.6	3.7	3.7	4.8
Mean	7.03	4.26	4.34	4.45
StDev.	2.204	2.458	0.844	0.663

(seconds)

each of the 12 locations. The circle inside the interquartile range represents the mean value of the data. The median values and interquartile ranges varied by locations. However, the median and mean values in total measured 1.08 and 1.05 degrees, respectively. These values correspond to about 1 cm at the center of the 3DSD. The range of quartiles is at most 1 degree or less, except at the sixth location. At the sixth location, pointing diagonally upward might be the cause of the increase in the differences. In practical use, most users are not aware of any differences within this range. The one-way Analysis of Variance (ANOVA) shows that there is a significant effect of the location on the pointing accuracy [$F(11, 1392) = 2.26, p < .01$].

B. *Pointing Stability*

In the previous calibration using the Motion Platform, the correlation between the IMU's yaw angle and the measured yaw angle was the lowest compared to the pitch and roll angles. In order to investigate how this result affects the stability of the pointing operation, we performed an experiment of pointing solely in the yaw direction.

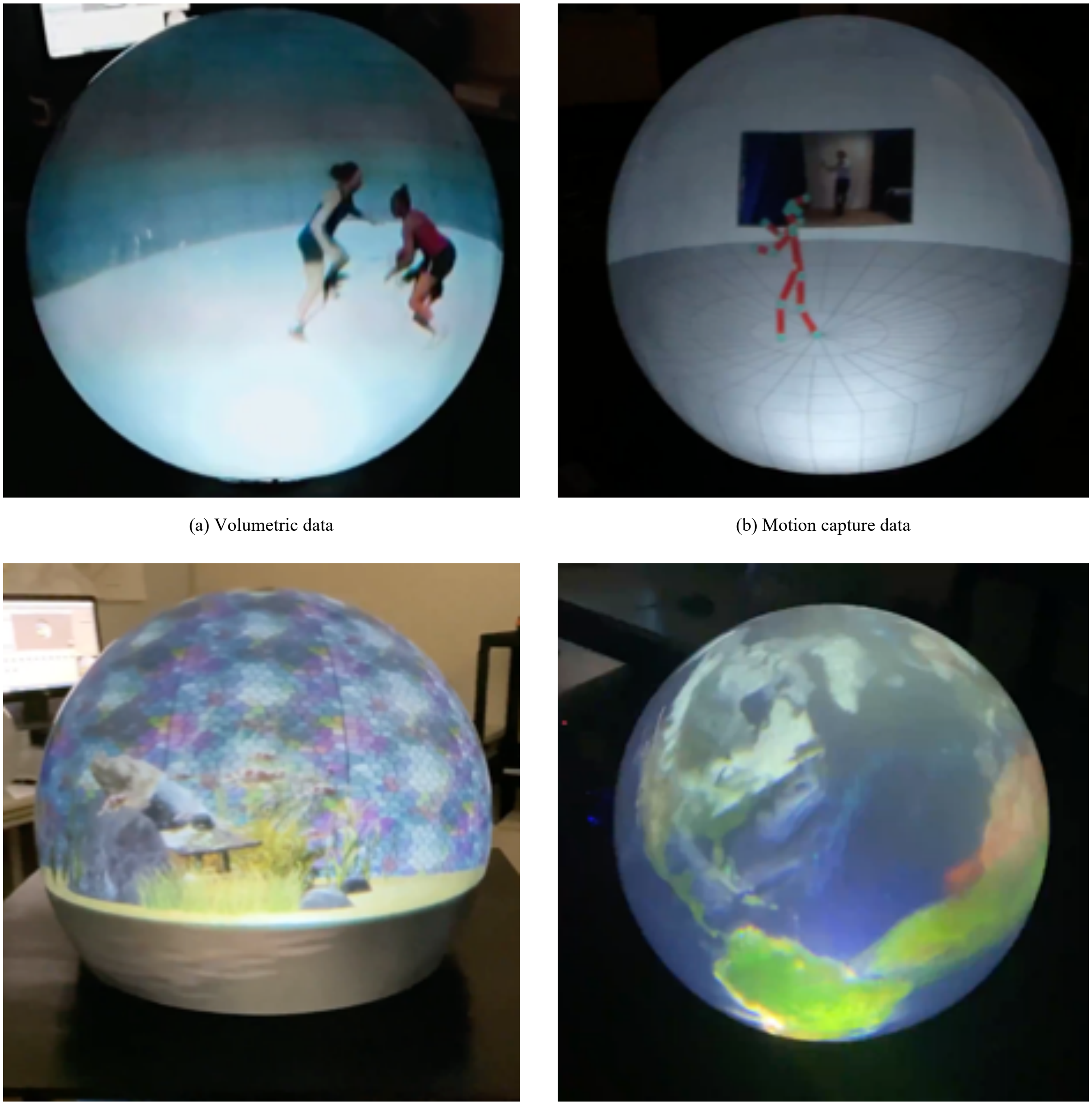
Figure 12 shows a schematic diagram of the experiment. The pointing device is placed on a board placed on the side of the 3DSD to allow pointing only in the yaw angle direction. Here, the pointing device was rotated ± 135 degrees with respect to the center of the 3DSD and the vectors representing the pointing direction were recorded. In total, 1,147 vectors were collected during the operation.

In order to verify the pointing stability, we calculated the intersection points (x, y, z) of these vectors with the 3DSD and fitted a plane $f(x, y) = z = ax + by + c$ to these points using the least squares method. The maximum absolute value of the residual was measured 0.20 cm. This small error means that the pointing vector changes smoothly in the yaw direction and the proposed pointing device is sufficiently stable. Figure 13 shows the distribution of the intersection points on the 3DSD.

C. *Object Selection*

We conducted an experiment to select objects in 3DSD using the proposed pointing device. For the experiment, we constructed a virtual fish-tank in 3DSD and placed four fishes at a distance from each other. We measured the time required to select every fish.

Nine students of the Faculty of Software and Information Science of Iwate Prefectural University participated in the experiment. Each subject randomly selected each fish using the proposed pointing device. However, in order to avoid gaps in results due to differences in pointing distance between



(a) Volumetric data

(b) Motion capture data

(c) Virtual fish-tank

(d) 3D digital globe

Figure 15. Various 3D contents that can be displayed onto the 3DSD.

subjects, we had all subjects start pointing at the same point. When the selection was successful, the color of the fish changed automatically and the time since the start of pointing was recorded. The experiment ended when all four fishes have been pointed out successfully.

Figure 14 shows the virtual fish-tank for the experiment. The "touch location" is a location to start to point to a fish, regardless of the location of the fish. The four fishes (A, B, C, D) are 34.9 cm, 22.6 cm, 23.7 cm, and 21.8 cm away from this point, respectively.

Table I shows the time being taken by each subject to select each fish. While the average time required to select fish in B, C, and D takes about 4 seconds, most subjects spent the most time to successfully select the fish A. We considered that it is a reasonable result since the fish A is the farthest away from the touch location. When selecting an object that is far away, a small amount of slippage in the pointing angle can reduce the pointing accuracy to the object. The one-way Analysis of Variance (ANOVA) shows that there is a significant effect between the pointing locations on the time taken by subjects to successfully point to a fish [$F(3, 32) = 4.46$, $p < .01$].

V. Discussion

Various spherical displays have been proposed in the past decade. The main purpose of developing these displays was to allow users to view content from various angles. To our knowledge, Sphere [15] is the first 2D spherical display that introduced multi-touch interactions, such as dragging, scaling, flicking, and rotating. This multi-touch function provides the same level of interaction as that of an ordinary planar display. However, as the use of 3D content increases, the direction of development of spherical displays has changed from 2D to 3D. Currently, there are some companies producing spherical displays, such as Global Imagination [16], PufferSphere [17], and ArcScience [18].

In 2011, SnowGlobe [6] was demonstrated to be the first 3DSD with a fish-tank VR. It has the same multi-touch functionality as the Sphere, allowing for interaction with 3D objects. However, interaction with 3D objects is likely to take advantage of the depth information, which requires a complex system to realize. Spheree [7] demonstrated 3D gestures using an optical motion capture device. Currently, CoGlobe [8] might be the most advanced 3DSD that allows two users to interact with the display at the same time. However, as long as the 3D interactions rely on expensive motion capture devices, the 3DSD may not become widely available. A 3D pen stylus that does not require a motion capture device such as the Touch™ Haptic Device [19] may be used to perform 3D interaction with the display within a certain range. However, its arm limits the work area.

This study extends the touch interface to enable 3D interactions without the use of a motion capture device. For this purpose, we have built a new 3DSD that has the same functionality as CoGlobe. Our proposed pointing device combines the 3D position on the display surface detected by the touch interface with the orientation measured by the IMU to obtain the 6DoF. Due to the high precision of recent IMUs, such as the one built into the iPhone 7, we believe that it is reasonable to obtain the orientation information of the pointing device from the IMU. Our calibration results using the Motion Platform indicate that the IMU built into the iPhone 7 is accurate enough to perform interactions that require absolute orientation.

The 3DSD has the potential to represent a variety of 3D contents, such as volumetric data [20], motion capture data, virtual fish-tank, and 3D digital globe, as shown in Figure 15. Since the proposed pointing device uses a mobile device, its functionality can be further enhanced by software application programs on the device. For example, after selecting a 3D object, gestures such as rotating and scaling can be performed on the mobile device screen to manipulate the 3D object. The capability to control the length of the auxiliary line from the tip of the pointing device may allow the user to sculpt or slice the 3D objects. Moreover, by using the haptic sensor inside the mobile device, the user can receive a haptic feedback when touching a 3D object.

VI. Conclusion

In this study, we have proposed a new pointing device that can be used to interact with a 3D spherical display without using a motion capture device. Our proposed device is characterized by the two different sources to calculate its 6DoF. They are a touch interface to acquire the location of the device tip and an IMU to measure the orientation of the device. We built a Motion Platform with three actuators to calibrate the IMU. Our experiments have confirmed the pointing accuracy and stability for the device.

The proposed pen has not yet been studied in detail in terms of resolution and sensitivity, but we will further experiment to clarify it in the future. We will also extend the functionality of the proposed pointing device as an interface in virtual surgical training based on a spherical display.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Numbers 20K12123. We also thank to the Iwate Prefectural University of Japan, for funding this project. We also thank the editor and three anonymous reviewers for their constructive comments, which helped us to improve the manuscript.

References

[1] R. Takahashi, K. Hotta, O. D. A. Prima, and H. Ito, “A Perspective-Corrected Stylus Pen for 3D Interaction,” The Thirteenth International Conference on Advances in Computer-Human Interactions (ACHI 2020), pp. 11-14, 2020.

[2] A. Borrego, J. Latorre, M. Alcañiz, and R. Llorens, “Comparison of Oculus Rift and HTC Vive: Feasibility for Virtual Reality-Based Exploration, Navigation, Exergaming, and Rehabilitation,” Games for Health Journal, 7(3), pp. 151-156, 2018.

[3] B. Masia, G. Wetzstein, P. Didyk, and D. Gutierrez, “A survey on computational displays: Pushing the Boundaries of Optics, Computation, and Perception,” Computers and Graphics (Pergamon), 37(8), pp. 1012-1038, 2013.

[4] Looking Glass, https://lookingglassfactory.com. [retrieved: September, 2020]

[5] J. Geng, “Three-Dimensional Display Technologies,” Advances in Optics and Photonics, 5(4), pp. 456–535, 2013.

[6] J. Bolton, K. Kim, and R. Vertegaal, “SnowGlobe: A Spherical Fish-Tank VR Display,” In Conference on Human Factors in Computing Systems - Proceedings, pp. 1159-1164, 2011.

[7] F. Ferreira et al., “Spheree: A 3D Perspective-Corrected Interactive Spherical Scalable Display,” ACM SIGGRAPH 2014 Emerging Technologies, pp. 1, 2014. https://doi.org/10.1145/2614217.2630585

[8] Q. Zhou et al., “CoGlobe - A Co-located Multi-Person FTVR Experience,” ACM SIGGRAPH 2018 Emerging Technologies. https://doi.org/10.1145/3214907.3214914.

[9] G. Hagemann, Q. Zhou, I. Stavness., O. D. A. Prima, and S. Fels, “Here’s looking at you: A Spherical FTVR Display for Realistic Eye-Contact,” ISS 2018 - Proceedings of the 2018

ACM International Conference on Interactive Surfaces and Spaces, pp. 357‑362, 2018.

[10] D. Fafard et al., "FTVR in VR: Evaluating 3D performance with a simulated volumetric fish-tank virtual reality display," Conference on Human Factors in Computing Systems, pp. 1-12, 2019.

[11] T. Crespel, P. Reuter, and X. Granier, "A Low-Cost Multitouch Spherical Display: Hardware and Software Design," Display Week 2017, May 2017, Los Angeles, California, United States. pp. 619- 622, 10.1002/sdtp.11716 . hal-01455523.

[12] OptiTrack, https://optitrack.com. [retrieved: September, 2020]

[13] P. C. Wu et al., "DodecaPen: Accurate 6DoF Tracking of a Passive Stylus," UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp. 365–374, 2017.

[14] Vive Tracker, https://www.vive.com/eu/vive-tracker/ [retrieved: September, 2020]

[15] H. Benko, A. D. Wilson, and R. Balakrishnan, "Sphere: Multi-Touch Interactions on a Spherical Display," UIST 2008 - Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology, pp. 77-86, 2008.

[16] S. W. Utt, P. C. Rubesin, and M. A. Foody, "Display System Having a Three-Dimensional Convex Display Surface," US Patent 7,352,340. 2005.

[17] Pufferfish Ltd. http://pufferfishdisplays.co.uk/, 2002. [retrieved: September, 2020]

[18] L. Thomas, F. Christopher, and L. Jonathan, "A Self-Contained Spherical Display System," In ACM Siggraph 2003 Emerging Technologies.

[19] The Touch™ Haptic Device, 3D Systems, https://www.3dsystems.com/haptics-devices/touch. [retrieved: September, 2020]

[20] T. Grossman, D. Wigdor, and R. Balakrishnan, "Multi-Finger Interaction with 3D Volumetric Displays," In UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology, pp. 61-70, 2004.