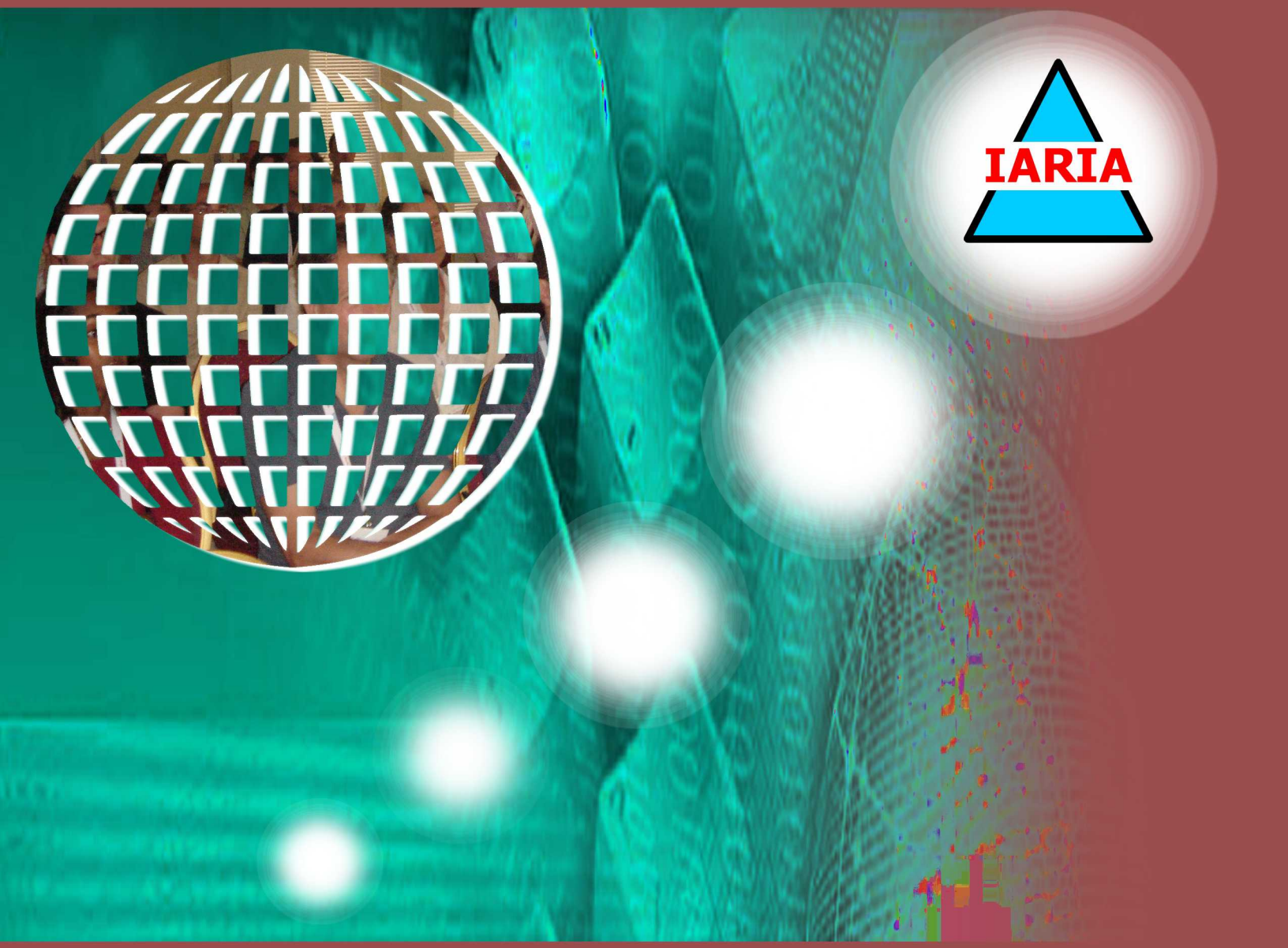
International Journal on Advances in Software



2013 vol. 6 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

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Software
Volume 6, Number 3 & 4, 2013

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
Mikey Browne, IBM, USA
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 del Este, 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
Alexandra Suzana Cernian, University POLITEHNICA of Bucharest, Romania
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
Hepu Deng, RMIT University, Australia
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
Lars Ebrecht, German Aerospace Center (DLR), Germany
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
Robert L. Glass, Griffith University, Australia
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, University of Ulm, Germany
Carlos Granell, European Commission / Joint Research Centre, Italy
Daniela Grigori, Université de Versailles, France
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
Phuong H. Ha, University of Tromso, Norway

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
Roger "Buzz" King, University of Colorado at Boulder, 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
Philippe Lahire, Université de Nice Sophia-Antipolis, France
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
Shuai Ma, Beihang University, China
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
Leonardo Mariani, University of Milano Bicocca, 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
Muhanna A Muhanna, University of Nevada - Reno, USA
Antonio Navarro Martín, Universidad Complutense de Madrid, Spain
Filippo Neri, University of Naples, Italy
Toàn Nguyên, INRIA Grenobel Rhone-Alpes/ Montbonnot, France
Muaz A. Niazi, Bahria University, Islamabad, Pakistan
Natalja Nikitina, KTH Royal Institute of Technology, Sweden
Marcellin Julius Nkenlifack, Université de Dschang, Cameroun
Roy Oberhauser, Aalen University, Germany
Pablo Oliveira Antonino, Fraunhofer IESE, Germany
Rocco Oliveto, University of Molise, Italy
Sascha Opletal, Universität Stuttgart, Germany
Flavio Oquendo, European University of Brittany/IRISA-UBS, France
Claus Pahl, Dublin City University, Ireland
Marcos Palacios, University of Oviedo, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Kai Pan, UNC Charlotte, USA
Yiannis Papadopoulos, University of Hull, UK
Andreas Papasalouros, University of the Aegean, Greece
Rodrigo Paredes, Universidad de Talca, Chile
Päivi Parviainen, VTT Technical Research Centre, Finland
João Pascoal Faria, Faculty of Engineering of University of Porto / INESC TEC, Portugal

Fabrizio Pastore, University of Milano - Bicocca, Italy
Kunal Patel, Ingenuity Systems, USA
Óscar Pereira, Instituto de Telecomunicacoes - University of Aveiro, Portugal
Willy Picard, Poznań University of Economics, Poland
Jose R. Pires Manso, University of Beira Interior, Portugal
Sören Pirk, Universität Konstanz, Germany
Meikel Poess, Oracle Corporation, USA
Thomas E. Potok, Oak Ridge National Laboratory, USA
Dilip K. Prasad, Nanyang Technological University, Singapore
Christian Prehofer, Fraunhofer-Einrichtung für Systeme der Kommunikationstechnik ESK, Germany
Ela Pustułka-Hunt, Bundesamt für Statistik, Neuchâtel, Switzerland
Mengyu Qiao, South Dakota School of Mines and Technology, USA
Kornelije Rabuzin, University of Zagreb, Croatia
J. Javier Rainer Granados, Universidad Politécnica de Madrid, Spain
Muthu Ramachandran, Leeds Metropolitan University, UK
Thurasamy Ramayah, Universiti Sains Malaysia, Malaysia
Prakash Ranganathan, University of North Dakota, USA
José Raúl Romero, University of Córdoba, Spain
Henrique Rebêlo, Federal University of Pernambuco, Brazil
Bernd Resch, Massachusetts Institute of Technology, USA
Hassan Reza, UND Aerospace, USA
Elvinia Riccobene, Università degli Studi di Milano, Italy
Daniel Riesco, Universidad Nacional de San Luis, Argentina
Mathieu Roche, LIRMM / CNRS / Univ. Montpellier 2, France
Aitor Rodríguez-Alsina, University Autonoma of Barcelona, Spain
José Rouillard, University of Lille, France
Siegfried Rouvrais, TELECOM Bretagne, France
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Djamel Sadok, Universidade Federal de Pernambuco, Brazil
Arun Saha, Fujitsu, USA
Ismael Sanz, Universitat Jaume I, Spain
M. Saravanan, Ericsson India Pvt. Ltd -Tamil Nadu, India
Idrissa Sarr, University of Cheikh Anta Diop, Dakar, Senegal / University of Quebec, Canada
Patrizia Scandurra, University of Bergamo, Italy
Giuseppe Scanniello, Università degli Studi della Basilicata, Italy
Daniel Schall, Vienna University of Technology, Austria
Rainer Schmidt, Austrian Institute of Technology, Austria
Cristina Seceleanu, Mälardalen University, Sweden
Sebastian Senge, TU Dortmund, Germany
Isabel Seruca, Universidade Portucalense - Porto, Portugal
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

Alin Stefanescu, University of Pitesti, Romania
Lena Strömbäck, SMHI, Sweden
Kenji Suzuki, The University of Chicago, USA
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
Ioan Toma, STI, Austria
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Peter Trapp, Ingolstadt, Germany
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
Jianwu Wang, San Diego Supercomputer Center / University of California, San Diego, USA
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Sebastian Wieczorek, SAP Research Center Darmstadt, 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
Qiang Zhu, The University of Michigan - Dearborn, USA

International Journal on Advances in Software
Volume 6, Numbers 3 & 4, 2013

CONTENTS

A Graph-Based Language for Direct Manipulation of Procedural Models

Wolfgang Thaller*, Ulrich Krispel†, René Zmugg*, Sven Havemann*, and Dieter W. Fellner*†‡

*Institute of Computer Graphics and Knowledge Visualization, Graz University of Technology, Graz, Austria
{w.thaller, r.zmugg, s.havemann}@cgv.tugraz.at

†Interactive Graphics Systems Group (GRIS), TU Darmstadt, Darmstadt, Germany
u.krispel@gris.tu-darmstadt.de

‡Fraunhofer IGD and TU Darmstadt, Darmstadt, Germany
d.fellner@igd.fraunhofer.de

***Abstract*—Creating 3D content requires a lot of expert knowledge and is often a very time consuming task. Procedural modeling can simplify this process for several application domains. However, creating procedural descriptions is still a complicated task. Graph based visual programming languages can ease the creation workflow; however, direct manipulation of procedural 3D content rather than of a visual program is desirable as it resembles established techniques in 3D modeling. In this paper, we present a dataflow language that features novel contributions towards direct interactive manipulation of procedural 3D models: We eliminate the need to manually program loops (via implicit handling of nested repetitions), we introduce partial reevaluation strategies for efficient execution, and we show the integration of stateful external libraries (scene graphs) into the dataflow model of the proposed language.**

Keywords-procedural modeling, dataflow graphs, loops, term graphs

I. Introduction

This is a revised and augmented version of "Implicit Nested Repetition in Dataflow for Procedural Modeling", which appeared in the Proceedings of The Third International Conference on Computational Logics, Algebras, Programming, Tools, and Benchmarking (COMPUTATION TOOLS 2012) [1].

Conventional 3D models consist of geometric information only, whereas a procedural model is represented by the operations used to create the geometry [2]. Complex man-made shapes exhibit great regularities for a number of reasons, from functionality over manufacturability to aesthetics and style. A procedural representation is therefore commonly perceived as most appropriate, but not so many 3D artists accept a code editor as user interface for 3D modeling, and only few of them are good programmers. Recently, dataflow graph based visual programming languages for 3D modeling have emerged [3], [4]. These languages facilitate a graphical editing paradigm, thus allowing to create programs without writing code. However, such languages are not always easier to read than a textual representation [5]. Therefore, the goal is a modeler that allows direct manipulation of procedural content on the concrete 3D model, without any knowledge of the underlying representation (code), while retaining the expressiveness of dataflow graph based methods.

In this paper, we present a term graph based language for procedural modeling with features that facilitate direct manipulation. First, we give an overview of related work in Section II. Then, we give a summary of the requirements for the language in Section III. Next, in Section IV the language is formally defined, and a compilation technique to embed such models in existing procedural modeling systems is described. Section V describes how the language can be applied to different modeling operations. Going beyond our previous work in [1], Section VI describes a method for incrementally reevaluating a procedural model expressed in our language in response to user interaction. We conclude with a discussion and some points of future research.

II. Related Work

Procedural modeling is an umbrella term for procedural descriptions in computer graphics. As a procedural description is basically just a computer program, there are many possibilities to express procedural content.

One category are general purpose programming languages with geometric libraries, for example C++ with *CGAL* [6] or the Generative Modeling Language (*GML*) [2] which utilizes a language similar to Adobe's PostScript [7]. *Processing* [8] is an open source programming language based on Java with a focus on computer programming within a visual context.

As many professional 3D modeling packages contain embedded scripting languages, these can be used to express procedural content. Some representatives are for example MEL script for Autodesk Maya [9] or RhinoScript for Rhinoceros [10].

Some domain specific languages have successfully been applied to express procedural content. For example, based on the work of Stiny et al. [11] who applied the concept of formal grammars (string replacements) to the domain of 2D shapes, Wonka et al. [12] introduced *split grammars* for automatic generation of architecture. These concepts have further been

extended by Mueller et al. [13] into CGA Shape, which is available as the commercial software package CityEngine [14] that allows procedural generation of buildings up to whole cities.

Visual Programming Languages (VPLs) allow to create and edit programs using a visual editing metaphor. Many VPLs are based on a dataflow paradigm [15]; the program is represented by a graph consisting of *nodes* (which represent operations) and *wires* along which streams of *tokens* are passed. Some examples in the context of procedural modeling are the procedural modeler Houdini [4] and the Grasshopper plugin for Rhinoceros [10], which both feature visual editors for dataflow graphs. Furthermore, the work of Patow et al. [16] has shown that shape grammars can also be represented as dataflow graphs. Such a representation also allows established interaction metaphors to be accessible for procedural modeling packages [17].

Term Graphs [18] arose as a development in the field of term rewriting. While term graphs are intuitively similar to dataflow graphs, there is no concept of a stream of tokens. Term graphs are a generalization of terms and expressions which makes explicit sharing of common subexpressions possible. Formally, we base our work on the definitions given in [19] rather than on any dataflow formalism.

III. Language Requirements

Dataflow languages have a number of properties that make them very desirable for interactive procedural modeling. They allow efficient partial reevaluation in order to interactively respond to "localized" changes, they are expressive enough to cover traditional domains of procedural modeling such as compass-and-ruler constructions and split-grammars, and they can be extended in various ways to support repeated structures/repeated operations.

We are currently researching direct manipulation based user interfaces for dataflow-based procedural modeling. This means that the dataflow graph itself is not visible to the user; instead, the user interacts with a concrete instance of the procedural model, i.e., a 3D model generated from a concrete set of parameter values. The basic usage paradigm is that the user selects objects in this 3D view and applies operations to them; these operations are added to the graph.

The goal of keeping the graph hidden during normal user interaction leads to additional requirements for the language that differ from traditional approaches.

A. Repetition

Loops should not be represented explicitly, i.e., loops should not be represented by an object that needs to be visualized so the user can interact with it directly. Operations should be implicitly repeated when they are applied to collections of objects.

It must be possible to deal with **nested repetitions** as part of this implicit repetition behaviour. Existing dataflow-based procedural modeling systems use a "stream-of-tokens" concept, i.e., a wire in the dataflow graph transports a linear stream of tokens that all get treated the same by subsequent operations. Nested structures are not preserved in this model.

When directly interacting with a 3D model, we expect the user to frequently zoom to details of the model. For example, consider a model of a building façade that consists of several stories, each of which contains several identical windows, which in turn contain several separate window panes. A user will zoom in to see a single window on their screen and then proceed to edit that archetypal window, for example by applying some operation to two neighbouring window panes of that same window. All operations in the modeling user interface should always behave consistently, independent of whether the user is editing a model consisting of just a single window, or one of many windows. In both cases, the system needs to remember that a collection of window panes belongs to a single window. Thus, flat token streams are not suited to direct-manipulation procedural modeling.

B. Failures

There are many modeling operations that do not always succeed, e.g., intersection operations between geometric objects. When applying volumetric split operations, a volume might become empty, rendering (almost) all further operations on that volume meaningless.

Often, these failures have only local effects on the model, so aborting the evaluation of the entire model is excessive; rather, we propagate errors only along the dependencies in the code graph — if its sources could not be calculated, an edge is not executed. In many cases, this is exactly the desired behaviour and allows to easily express simple conditional behaviours such as "if there is an intersection, construct this object at the intersection point" or "if there is enough space available, construct an object".

C. Side Effects

Neither dataflow graphs nor term graphs are particularly well-suited for dealing with side-effecting operations; also, to simplify analysing the code for purposes of the GUI, we have a strong motivation to forbid side effects.

However, it is a fundamental user expectation to be able to have operations that *create* objects, and to be able to *replace* or refine objects. Both Grasshopper and Houdini use side-effect free operations and rely on the user to pick one or more dataflow graph nodes whose results are to be used for the final model; this solution is not applicable to a direct manipulation procedural modeler because it would require interacting with the graph rather than with a 3D model.

IV. The Language

Below, we will first define the term graphs that form the basis of our language; we will then proceed to discuss our treatment of side effects, repetition and failing operations.

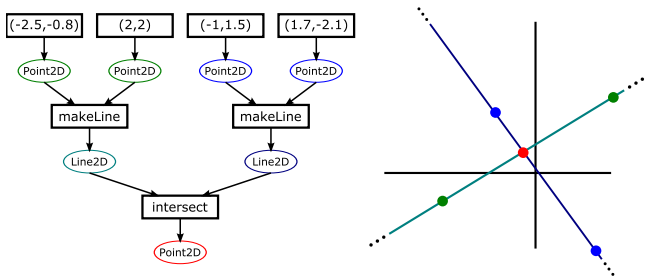


Fig. 1. A **code graph** (as presented by [19]) is a hypergraph that consists of nodes that correspond to results and hyperedges that represent operations (left). In this illustration the nodes are represented as ellipses. Hyperedges are visualized as boxes; they can have any number of source and target nodes. Hyperedges with no source nodes correspond to constants. This example shows a code graph that carries out a simple construction: Two points define a straight line; two lines yield an intersection point (right).

A. Code Graphs

The underlying data structure is a *hypergraph* consisting of nodes, which correspond to (intermediate) values and graphical objects, and *hyperedges*, which represent the operations applied to those values as shown in Figure 1.

Note that we are following term graph terminology here, which differs from the terminology traditionally used for dataflow graphs. In a dataflow graph, *nodes* are labelled with operations, and they are connected with edges or *wires*, which transport values or tokens. In a term graph, *hyperedges* (i.e., edges that may connect more or fewer than two nodes) are labelled with operations or literal constants, and values are stored in nodes, which are labelled with a type.

We reuse the following definition from [19]:

Definition 1: A *code graph* over an edge label set ELab and a set of types NType is defined as a tuple $G = (\mathcal{N}, \mathcal{E}, \mathsf{In}, \mathsf{Out}, \mathsf{src}, \mathsf{trg}, \mathsf{nType}, \mathsf{eLab})$ that consists of:

- a set $\mathcal{N}$ of *nodes* and a set $\mathcal{E}$ of *hyperedges* (or *edges*),
- two node sequences $\mathsf{In}, \mathsf{Out} : \mathcal{N}^*$ containing the *input nodes* and *output nodes* of the code graph,
- two functions $\mathsf{src}, \mathsf{trg} : \mathcal{E} \to \mathcal{N}^*$ assigning each edge the sequence of its *source nodes* and *target nodes* respectively,
- a function $\mathsf{nType} : \mathcal{N} \to \mathsf{NType}$ assigning each node its *type*, and
- a function $\mathsf{eLab} : \mathcal{E} \to \mathsf{ELab}$ assigning each edge its *edge label*. □

Furthermore, we require all code graphs in our system to be acyclic and that every node occurs exactly once in either the input list of the graph, or in exactly one target list of an edge.

Definition 2: Edge labels are associated with an input type sequence and an output type sequence by the functions $\mathsf{edgeInType}$ and $\mathsf{edgeOutType} : \mathsf{ELab} \to \mathsf{NType}^*$. □

Definition 3: An edge e is considered *type-correct* if $\mathsf{edgeInType}(\mathsf{eLab}(e))$ matches the type of the edge's source nodes, and $\mathsf{edgeOutType}(\mathsf{eLab}(e))$ matches the type of its target nodes. A codegraph is type-correct if all edges are type-correct. □

B. Limited Side Effects

In Section III-C, we have noted the need to be able to model *creation* and *replacement* operations. The *scene* is the set of visible objects; we define it as a global mutable set of object references. We only allow two kinds of side-effecting operations: (a) adding a newly-created object to the scene, thus making it visible; and (b) removing a given object reference from the scene.

Replacement and refinement can be modeled by removing an existing object and adding a new one. Object removal is idempotent and only affects object visibility, not the actual object. Object visibility cannot be observed by operations. Therefore, no additional constraints on the order of execution are introduced.

C. Implicit Repetition

When an operation is applied to a list rather than a single value, it is implicitly repeated for all values in the list; if two or more lists are given, the operation is automatically applied to corresponding elements of the lists (cf. Figure 2). It is assumed that the lists have been arranged properly.

We define our method of implicitly handling repetition by defining a translation from codegraphs with implicitly-repeated operations to codegraphs with explicit loops.

1) Explicit Loops:

Definition 4: A *codegraph with explicit loops* is a codegraph where the set of possible edge labels ELab has been extended to include *loop-boxes*. A loop-box edge label is a tuple $(\texttt{LOOP}, G', f)$ where G' is a code graph (the loop body) with n inputs and $f \in \{0,1\}^n$ is a sequence of boolean flags, such that at least one element of f is 1. The intention behind the flags f is to indicate which inputs are lists that are iterated over ($f_i = 1$), and which inputs are non-varying values that are used by the loop ($f_i = 0$). The number of iterations corresponds to the length of the shortest input list. The edge input and output types of a loop are defined by wrapping the input and output types of the loop body (referred to as ti_i and to_i below) with $\texttt{List}[\cdots]$ as appropriate:

$$\mathsf{edgeOutType}((\texttt{LOOP}, G', f))_i := \texttt{List}[to_i]$$

$$\mathsf{edgeInType}\left((\texttt{LOOP}, G', f)\right)_i := \begin{cases} \texttt{List}[ti_i] \text{ if } f_i = 1 \\ ti_i \text{ otherwise} \end{cases}$$ □

2) Codegraphs with Implicit Repetition: To allow implicit repetition, we relax the type-correctness requirement that edge input/output types match the corresponding node types.

A codegraph with implicit repetition is translated to a codegraph with explicit loops by repeatedly applying the following translation; the original codegraph is considered type-correct iff this algorithm yields a codegraph with explicit loops that fulfills the type-correctness requirement.

Consider an edge e where the type-correctness condition is violated. If any of the output nodes is not a list, or if any of the mis-matching input nodes is not a list, abort; in this case, the input codegraph is considered to be invalid. Replace the edge e by a loop edge e'. The repetition flags f_i for the new loop edge are set to 1 for every input with a type mismatch, and

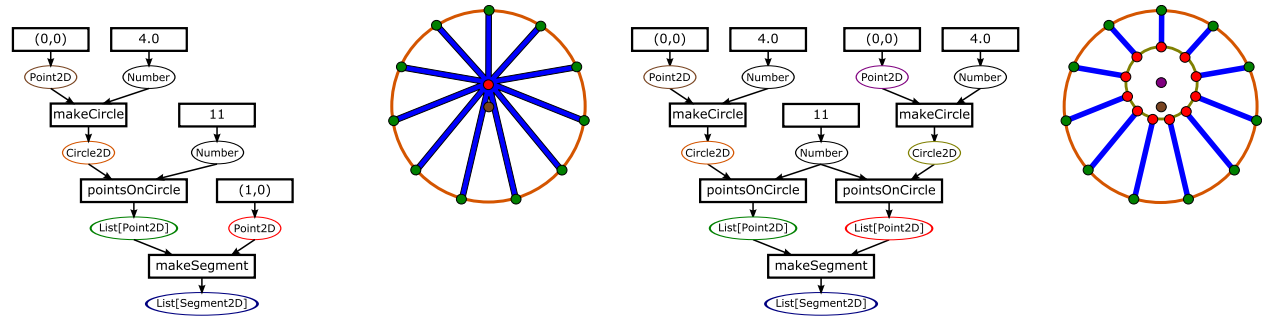


Fig. 2. Handling repetitions: The images show examples of simple procedural models ((b) and (d)) that create a list of line segments (blue) and their respective code graphs ((a) and (c)). Points, lines and circles correspond to intermediate results (nodes) of the same color. makeCircle creates a circle out of a point and a radius, pointsOnCircle creates a list of evenly distributed points on a circle and makeSegment creates a straight line segment between two points. This operation can be implicitly repeated to create segments from a list of points (on a circle) to a single point ((b)), or between two lists of points on circles ((d)) using makeSegment. Multiple graphical elements are represented by single nodes in the corresponding code graphs ((a) and (c)).

to 0 otherwise. The loop body G' is a codegraph containing just the edge e; the types of its input and output nodes are chosen such that the edge e' becomes type-correct within the outer codegraph. The translation is then applied to the loop body G'.

3) Fusing Loops: The result of the above translation is a codegraph that contains separate (and possibly nested) loops for each edge. This is undesirable for two reasons, namely performance and code readability. Performance is relevant whenever the operations used in the codegraph edges are relatively cheap, such as, for example, compass and ruler constructions, as opposed to boolean operations on 3D volumes (constructive solid geometry, CSG). Code readability is important because a procedural model might still need to be modified after it has been exported from our system to a traditional script-based system.

Consecutive loops, i.e., loops where the second loop iterates over an output of the first, can be fused if both loops have the same number of iterations and if the second loop does not, either directly nor indirectly, depend on values from other iterations of the first loop.

To determine which loops have the same number of iterations, we will annotate each occurrence of `List` in each node type with a symbolic item count, represented by a set of variable names. Each variable is an arbitrary name for an integer that is unknown at compile time. A set denotes the minimum of all the contained variables. $\texttt{List}_{\{a\}}[t]$ means a list of a items of type t, and $\texttt{List}_{\{a,b\}}[t]$ means a list of $\min(a, b)$ items.

All `List` types that appear as outputs of non-loop edges are annotated with a single unique variable name each. Every loop edge is annotated with a symbolic iteration count that is the minimum (represented by set union) of the symbolic item counts of all the lists it iterates over. Annotations on nested `List` types are propagated into and out of the loop bodies. The resulting `List` types of a loop box are annotated with a symbolic item count that is equal to the symbolic iteration count of the loop.

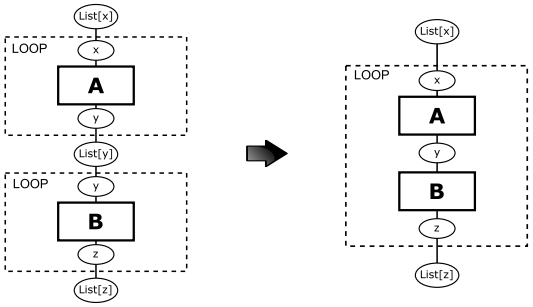


Fig. 3. Two consecutive loops containing one operation each that gets applied to every item of the list. Under certain conditions (see text) the loops can be fused in order to simplify the graph.

Two consecutive loop edges e_1 and e_2 can be *fused* when the symbolic iteration counts of the loops are equal, the repetition flag f_i is set to 1 for all inputs of e_2 that are outputs of e_1, and e_2 is not reachable from any edge that is reachable from e_1, other than e_1 and e_2 themselves.

If all these conditions are fulfilled for a given pair of edges, the edges can then be replaced by a single edge (cf. Figure 3); the fused loop body is the sequential concatenation of the two individual loop bodies. The inputs for the fused edge are the inputs of e_1 and all nodes that are inputs of e_2 but not outputs of e_1. The flags f_i for the fused edge are equal to the corresponding flags for inputs of e_1 and e_2. The outputs for the fused edge are all nodes that are either outputs of e_1 or of e_2. This fusing operation is applied until no more edges can be fused.

D. Handling Errors

The desired error-handling behaviour can be described by regarding `ERROR` as a special value which is propagated through the codegraph. If an operation fails, all its outputs are set to `ERROR`; an operation is also considered to fail whenever any of its inputs are `ERROR`.

In a naive translation, all arguments need to be explicitly checked for every single operation. To arrive at a better

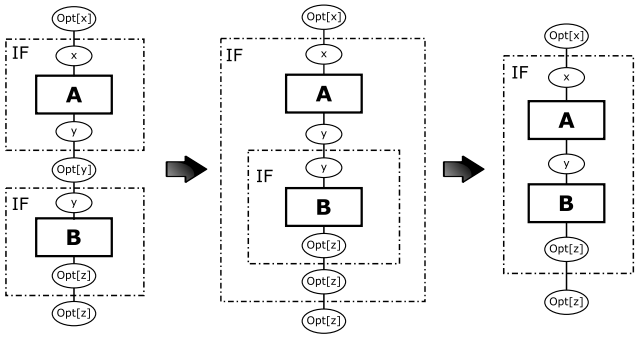
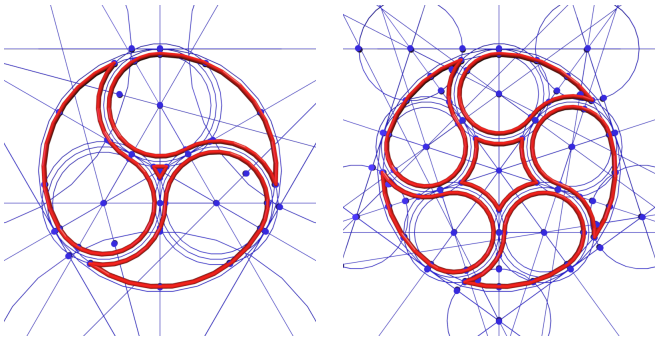


Fig. 4. Left: two consecutive if-boxes used for handling potentially-failing operations. The input (`Opt`$[x]$ at the top) is already the result of a potentially-failing operation. Note that in this example, operation **A** itself cannot fail (result type is plain y), while operation **B** can (result type is `Opt`$[z]$). They can be combined by nesting the second box inside the first (center). This often exposes opportunities for eliminating redundant error checks (right).



(a) (b)

Fig. 5. This gothic window construction was created in our test framework using direct manipulation without any code or graph editing. The number of repetitions is an input parameter of the model.

translation, we use a similar method as for the loops above; we first make the error checking explicit and then introduce a rule for combining consecutive error-checks.

Definition 5: `Opt`$[t] := t \cup \{$`ERROR`$\}$ for all types t, i.e., `Opt`$[t]$ is a type that can take any value that type t can, or a special error token. `Opt`$[t]$ is idempotent: `Opt`$[$`Opt`$[t]] =$ `Opt`$[t]$. Also note that `Opt` can nest with `List` — the types `Opt`$[$`List`$[t]]$ and `List`$[$`Opt`$[t]]$ and `Opt`$[$`List`$[$`Opt`$[t]]]$ are three different types. □

Definition 6: An *if-box* edge label is a tuple $(\texttt{IF}, G', f)$ where G' is a codegraph with n inputs and $f \in \{0,1\}^n$ is a sequence of boolean flags, such that at least one element of f is 1. The edge input and output types of a loop are defined by wrapping the input and output types of the loop body with `Opt`$[\cdots]$ as appropriate, analogously to the treatment of loop boxes (cf. Definition 4). When an if-box is executed, all input values for which $f_i = 1$ are first checked for `ERROR`s; if any of the input values is equal to `ERROR`, execution of the box immediately finishes with a result value of `ERROR` for each output. If none of the inputs are `ERROR`, the body G' is executed; its output values are the output values of the if-box. □

Predefined operations that can fail will return optional values (`Opt`$[\cdots]$). For every edge in the code graph, if-boxes have to be inserted if necessary to make the codegraph type-consistent.

Two consecutive if-box edges e_1 and e_2 can be *fused* when the flag f_i is set to 1 for at least one input e_2 that is an output of e_1, and e_2 is not reachable from any edge that is reachable from e_1, other than e_1 and e_2 themselves.

Fusing of if-boxes happens by moving the edge e_2 into the body of the if-box e_1, yielding two nested if-boxes (cf. Figure 4). The inputs for the fused edge are the inputs of e_1 and additionally all nodes that are inputs of e_2 but not outputs of e_1; the flags f_i for the additional flags are all set to 0, which means that the outer box does not need to check these inputs against `ERROR`, because the inner box will do so if necessary. For the nested if-box inside the fused edge, we next check whether that box is still required; first, for every input whose node type is not of the form `Opt`$[t]$, the corresponding flag f_i is set to 0. If all flags are set to zero for the inner if-box, the box is eliminated by replacing the edge with its body codegraph.

V. Modeling Vocabularies

In this section, we describe some common modeling operations and their realization within our framework. The examples in this section have been created using direct manipulation on a visible model only (without visualization of the underlying code graph), the concrete user interface is, however, outside the scope of this paper. Refer to [20] and [21] for accounts of different applications of our system.

A. Compass & Ruler

Compass and ruler operations have long been used in interactive procedural modeling [22]; these operations are well suited to a side-effect free implementation, and usually return only a single result per operation. Our addition of repetition allows for new constructions (Figure 5).

B. Split Grammars

We can use a methodology similar to Patow et al. [16] to map split grammars to code graphs (see Figure 6). A model is described by a set of replacement rules. Successive application of rules gradually refines the result (coarse to fine description). Just as in CGA Shape [13] and the work of Thaller et al. [23], a shape consists of a bounding volume called *scope*, a individual local coordinate system, and geometry within the scope. These shapes are partitioned into smaller volumes by operations split and repeat (replacement as side-effect). The split operation partitions the scope in a predefined number of parts, whereas with the repeat operation the number of parts is determined by the size of the scope at the time of rule application.

A complex example of a façade that was realized through our system is shown in Figure 7.

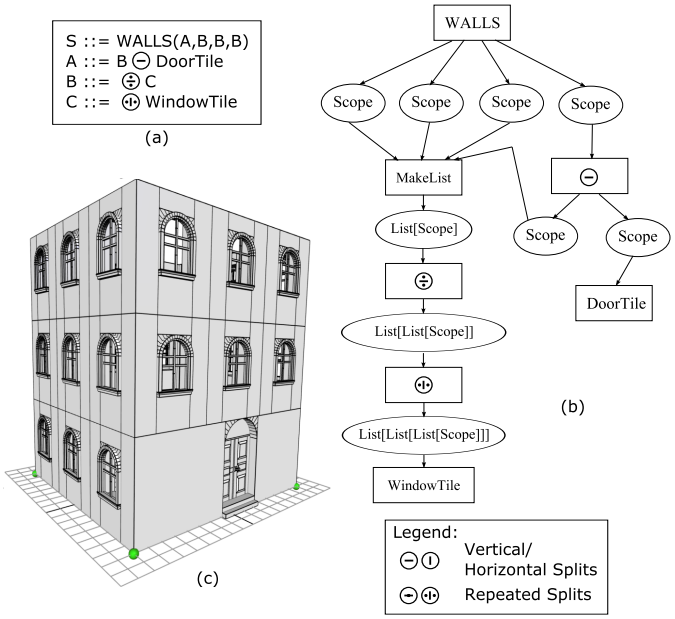
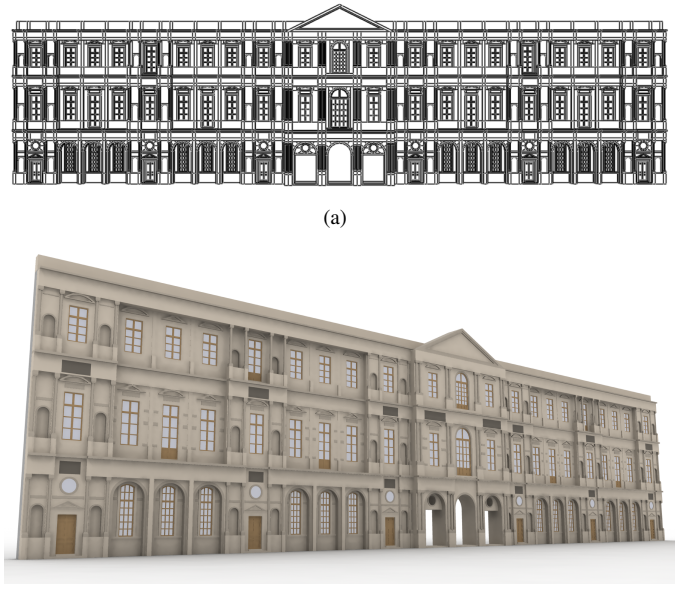


Fig. 6. Split grammar example: A simple shape grammar with split and repeat operations can be expressed using a textual description (a). This structure can be mapped to a codegraph (b) and executed, which yields (c).



(a)

(b)

Fig. 7. A complex façade example realized with our system. These images stem from parts of the Louvre that were reconstructed in the work of Zmugg et al. [20]. Figure (a) shows the hierarchical split layout of the façade, which led to the rendering (b).

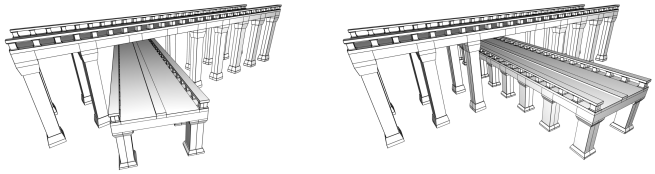


Fig. 8. Illustration of interconnected structures: The pillars of the bridges are constructed using ray casting for obstacle detection; The pillars of the larger bridge are constructed with respect to the position of the lower bridge.

C. Interconnected Structures

A drawback of (context-free) shape grammar systems is that they lack mechanisms for connecting structures across different parts of the top-down modeling hierarchy. The solution proposed in [24] is to extend a text-based shape grammar system by a feature called "containers". The idea is to pass *mutable* containers, currently implemented via nested arrays, as parameters to shape grammar rules. During the evaluation of the rules, objects which are potential *attaching points* can be appended to these arrays as a side effect of the grammar evaluation. These arrays can later be used to define structures that connect elements from independent parts of the model hierarchy. These connecting elements can follow different connection patterns based on geometric queries, such as ray casting (see Figure 8).

We can directly translate the idea of containers in [24] to our system, with the only difference being that attaching points have to be explicitly grouped in arrays. This does not cause additional complexity; receiving a container as an input and explicitly adding objects should take about the same "effort" as explicitly grouping objects and returning a nested list as an output. In the context of direct manipulation of procedural models, however, our approach has two advantages, both of which stem from the absence of side effects in our system:

- A list in our system has a concrete visual representation — the user can click on it; by contrast, a *mutable* container has different states throughout its lifetime, and it is created as an empty container before objects are added. As such it is a "virtual" object for which no concrete 3D representation seems possible.
- Passing a mutable container enforces a linear execution order; different operations that access the same container must always be executed in the same order, or the meaning will change, preventing efficient partial reevaluation of the scene. This is not a problem in the context of [24], which focuses on offline generation of geometry.

D. Scene Graphs

Many three-dimensional scenes have a hierarchical structure where individual objects are placed relative to a parent object, e.g., pieces of furniture are placed relative to the room that they are located in, but the objects on a table are placed relative to the table. The structure describing such relations is referred to as a *scene graph*. When objects that occur more

Fig. 9. Scene graphs allow the representation of hierarchical dependencies; As the TV is placed in dependence of the table, changing the table's position will also move the TV accordingly. Furthermore, scene graphs are memory efficient through instancing: the two chairs in this scene refer to the same geometry with different transformations.

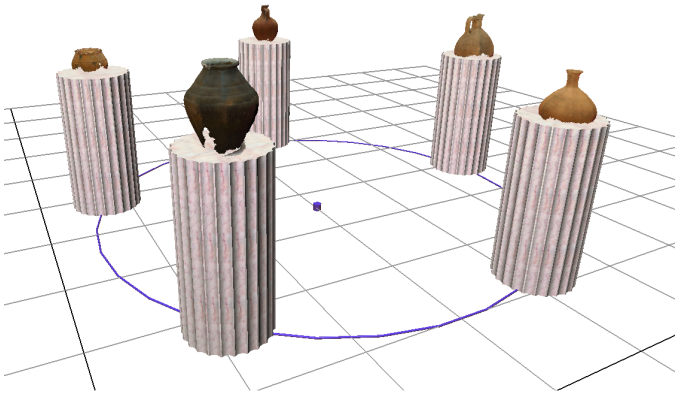


Fig. 10. A procedural scene graph: Scene graph nodes with pedestals are placed at points distributed on a circle. On top of each pedestal, a museum exhibit is placed. The input for the code graph that represents this procedural model is a list of file paths to load the exhibits from.

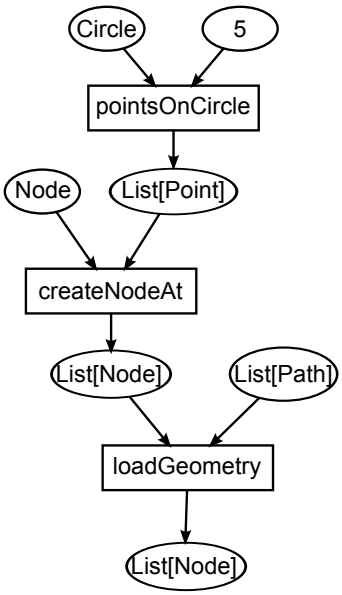


Fig. 11. The code graph representing Figure 10. For simplicity, the operations representing the pedestals have been left out.

than once in a scene are taken into account, the scene graph is an acyclic graph instead of a tree. Each node in a scene graph contains a transformation and, optionally, geometry. The transformation that is applied to each piece of geometry — defining its placement in the scene — is the product of all transformations on the path from the root to the node (see Figure 9). By gradually changing the transformations of nodes, animated objects can be achieved easily.

To embed a scene graph in our system, we first need a type `Node` to represent scene graph nodes; we will assume that one instance of that type, the *root* of the scene graph, will get passed to the code graph as an input. Child nodes are created using operations that take the parent as an input; in particular, there is an operation createNode that creates a transformation node (without any visible geometry) and an operation loadGeometry that creates a node with pre-generated geometry loaded from a file. Finally, the toGlobal converts a point from the local coordinate system of a scene graph node to global coordinates; this operation allows creating structures that connect objects that reside in different parts of the scene graph.

Building on top of the createNode and toGlobal operations, we can also provide a createNodeAt operation that places a child node at a point given in global coordinates (instead of the usual parent-relative transformation).

The first question that has to be asked is whether this vocabulary fulfills the requirements of our code graph formalism, in particular the limitations on side effects. There are no object removal or replacement operations; both createNode and loadGeometry are intended as object creation functions, but they actually modify the parent node's set of children rather than a global set of visible objects. This is, however, equivalent to having one global set of objects, where each object can contain a reference to its parent object. Storing individual sets of child nodes at each node rather than a single global set can thus be seen as a performance optimisation that is transparent from the semantics point of view.

For real-world applications, the range of node-creating operations needs to be extended, but the basic structure will remain the same. This is a straightforward mapping of a scene graph to the code graph, which is important because it allows the user interface to present standard scene graph semantics to the user. The work of Zmugg et al. [21] describes this from an application's point of view.

Figure 10 shows a variant of a use case described in [21]; the corresponding code graph can be seen in Figure 11. The inputs for this code graph are the number of museum exhibits to be shown and a list of paths to files containing the 3D models of the exhibits. The requested number of models is loaded from the list and placed in scene graph nodes arranged in a circle.

As the transformations could be represented explicitly as values in the code graph, a code graph based system is necessarily at least as powerful as a scene graph system.

However, there are two reasons why it is desirable to use existing scene graph systems (such as OpenSG [25]) as a modeling vocabulary for a code graph based system:

- Scene graphs, with their hierarchical way of managing object placement, provide a useful abstraction; dealing with transformations as values in a code graph can be hard to understand for the end user. A scene graph node, on the other hand, ties the transformation to a concrete object and can thus be represented in a more intuitive way in a graphical user interface.
- Scene graph systems are available as ready-to-use libraries and already solve many problems related to efficient rendering and animation. It is therefore desirable to be able to use them from the procedural modeling system, rather than having to re-implement their functionality.

VI. Incremental Updates

During interactive manipulation of a procedural object, it is usually a single parameter that is being modified, for example using a mouse dragging operation. This parameter often only affects a small part of the model, so, for reasons of performance, it is not desirable to reevaluate the entire model. Instead, we want to perform the minimum amount of work required, i.e., to only reevaluate those individual operations that really depend on the changed input.

The straightforward method is thus to reevaluate all hyperedges that are below the changed value (i.e., that consume it directly or indirectly). Reevaluating an operation entails first undoing all side effects caused by that operation, before re-executing the operations with updated parameters.

In the course of this section, we will first show why this approach is not sufficient and then proceed to describe a method that takes the discussed problems into account.

A. The Problem of Aggregate Values

Excessive recalculation can happen whenever an input of a code graph edge is of an aggregate type, i.e., any type that consists of several parts such that some of these parts might stay unchanged. Lists are an obvious example of an aggregate data type in our language; if a change in an input parameter causes a change in one element of an intermediary result of list type, we do not want to undo and recalculate all operations that use the unchanged elements.

Individual modeling vocabularies can add further aggregate data types which can also cause excessive recalculation; changing the color of an object might, for example, only affect the colors of objects that depend on it, but not their shape. Recalculating the geometry of those objects might be a lot more expensive than just updating their color.

In particular, this problem affects our use of scene graphs as described in Section V-D. One of the main strengths of scene graphs is that a scene graph can be efficiently animated by changing the transformation on a node; this affects all children of the node without requiring that subgraph to be changed. The code graph representation of a scene graph, however, encodes dependencies that do not actually exist. Each scene graph node depends on its transformation, and all children of a scene graph node depend on the parent node.

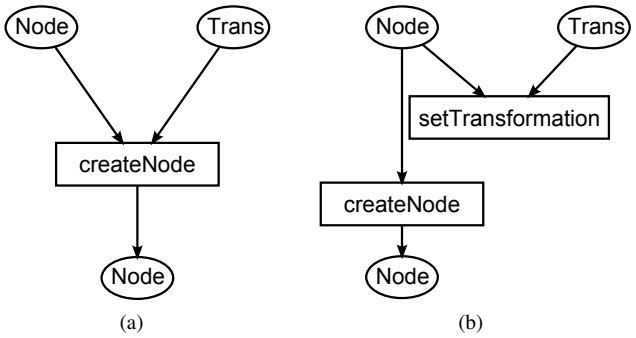


Fig. 12. With a simple representation of scene graphs, child nodes will depend on the transformation (a); this can be avoided by having separate createNode and setTransformation operations (b). This will avoid a reevaluation of createNode after changing the transformation.

Thus, when a naive method is used to evaluate the code-graph, all children of a node are rebuilt from scratch when the transformation of the parent is changed.

The "museum" example from Figures 10 and 11 constitutes a further example. When the number of museum exhibits to be displayed is changed, already-loaded objects should never be re-loaded. Ideally, only new objects should be loaded, while all objects that were previously visible should be re-used.

B. A Possible Alternate Interface to Scene Graphs

A possible way to solve this problem is to make the code graph encode the dependencies more accurately. In particular, this means using several separate operations to achieve the work of createNode and related operations. In particular, node creation needs to be separated from setting a node's transformation (Figure 12). A node's children will thus depend on the parent node's existence, but not on its transformation.

This hypothetical setTransformation instruction introduces some problems. It is obviously a side-effecting operation, and it does not seem to fit any of the allowed side effects described in Section IV-B. It can, however, be interpreted as an object creation function that creates an invisible "transformation object" that contains the transformation and a reference to the scene graph node to be transformed. After code graph evaluation is complete, the scene can be scanned for these transformation objects and the transformations applied to the scene graph nodes.

The toGlobal operation can not be supported directly by this approach, as it would need to observe the set of transformation objects that are part of the scene, which is not allowed according to Section IV-B. Instead, the actual transformation will need to be passed to it using an explicit connection in the code graph.

This approach thus fulfills all the formal criteria, but it has serious disadvantages for the user interface. Direct manipulation of the setTransformation operation by itself is next to impossible, as it does not have any result that can be

visualized in a 3D GUI. The GUI layer will probably have to present a simplified view of the situation to the user, where the createNode and setTransformation operations are folded back together.

Likewise, note that the loadGeometry operation used in the example in Figure 11 is responsible for both loading a model from a file and placing it in the scene. Separating a loading operation from a placement operation would introduce additional complexity at the user interface level, as the user interface would need to have a visual representation of "loaded but not displayed" models. It is thus highly desirable to have a single operation for loading and placement.

It is therefore preferable by far to define a method to efficiently handle incremental updates on codegraphs that use the more straightforward representation of scene graphs described earlier.

C. Incremental Update for Individual Operations

To define the semantics of incremental updates, we will follow a bottom-up approach. We will start by defining updates for individual hyperedges in a code graph, i.e., for individual operations, before building up to entire code graphs.

Modeling operations are defined outside the code graph. When basic term graph evaluation is used to evaluate code graphs, each modeling operation can be implemented by a simple function in the underlying language of the system.

To partially reevaluate a code graph, the side effects of the affected operations have to be taken into account; in particular, old side effects have to be undone before an operation can again be executed with new parameters. Therefore, for each modeling operation op we require a modeling operation library to provide at least the two implementation functions $\mathsf{evaluate}_{op}$ and undo_{op}. The former performs the operation (potentially causing side effects) and the latter reverses those side effects.

$$\begin{aligned} \mathsf{evaluate}_{op} &: (in_1 \cdots in_n) \\ &\rightarrow (\mathtt{S}_{op}, out_1 \cdots out_m) \\ \mathsf{undo}_{op} &: \mathtt{S}_{op} \rightarrow () \end{aligned}$$

In addition to the modeling operation's output values, the $\mathsf{evaluate}_{op}$ function also returns a value that gets passed to the next invocation of undo_{op}. The type $\mathtt{S}_{op}$ of this state value depends on the operation. For operations without side effects, the undo_{op} function does nothing and $\mathtt{S}_{op}$ contains no information.

To allow reevaluations to be optimized by taking advantage of previous evaluations and to allow partial updates of aggregate values such as lists and scene graph nodes, modeling operation libraries may also provide a update_{op} function.

The input and output values of the update_{op} function will each be annotated with a tag. A tag is intended to specify if a value has changed, and if so, in what way it has changed. For many datatypes, it will be enough to track whether a value has changed or not. For scene graph nodes, we need to distinguish between changes that only affect the transformation and changes that should trigger a complete reevaluation of subsequent operations.

We therefore define a parametric datatype Tag, i.e., for each data type t, we define a datatype $\mathtt{Tag}[t]$. The individual Tag datatypes should be defined individually according to the requirements for the datatype t. We require that every Tag datatype has at least the two special values NEW and UNCHANGED.

The update_{op} function is also supplied with the outputs of the previous evaluation of the operation; its signature is thus as follows:

$$\begin{aligned} \mathsf{update}_{op} &: (\mathtt{S}_{op}, in_1 \cdots in_n, \\ &\qquad \mathtt{Tag}[in_1] \cdots \mathtt{Tag}[in_n], \\ &\qquad oldout_1, \cdots oldout_m) \\ &\rightarrow (\mathtt{S}_{op}, out_1 \cdots out_m, \\ &\qquad \mathtt{Tag}[out_1] \cdots \mathtt{Tag}[out_m]) \end{aligned}$$

If the update function is not defined, a default implementation is provided based on the evaluate and undo functions; it can be seen in Listing 1.

Listing 1 Default update function for operations

function $\mathsf{update}_{op}(s,\ arg_{1\cdots n},\ atag_{1\cdots n},\ oldout_{1\cdots m})$
  if all $atag_i$ are UNCHANGED **then**
    return $(s,\ oldout_{1\cdots m},$
      UNCHANGED $\cdots$ UNCHANGED)
  else
    $\mathsf{undo}_{op}(s)$
    $(s',\ out_1 \cdots out_m) \leftarrow \mathsf{evaluate}_{op}(arg_0 \cdots arg_n)$
    return $(s',\ out_{1\cdots m},$ NEW $\cdots$ NEW)
  end if
end function

To solve the problem of transformations causing subsequent scene graph nodes to be recreated, we need to define tags for scene graph nodes that can describe the situation that only the transformation has changed:

$$\mathtt{Tag}[\mathtt{Node}] := \{\mathtt{NEW}, \mathtt{UNCHANGED}, \mathtt{TRANSFORMED}\}$$

For scene graph nodes that are TRANSFORMED, most update operations will simply reuse their old outputs, but tag any scene node outputs as TRANSFORMED as well; the toGlobal operation, however, which actually depends on the transformation of the input, will calculate a new result and tag it as NEW.

D. Incremental Update for Entire Code Graphs

We are now ready to define $\mathsf{evaluate}_G$, undo_G and update_G on entire codegraphs.

For the purpose of this section, we assume G to be a code graph without implicit repetition or error handling.

Definition 7: For each code graph G, we define $\mathsf{evaluate}_G$ to be the evaluation function on the entire code graph. Evaluating a code graph entails calling the individual $\mathsf{evaluate}_{op}$

functions for each edge of G in an arbitrary topologically sorted order. The state value returned is a map associating each edge of G with the state value returned by the $\mathsf{evaluate}_{op}$ invocation for that edge. □

Definition 8: The undo function undo_G on a code graph G calls the individual undo_{op} for all the edges of G in an unspecified order. Each individual undo_{op} is passed the appropriate state value. □

Definition 9: The function update_G calls each update_{op} function for each edge of G in an arbitrary topologically sorted order. Each individual update_{op} is passed the appropriate state value from the input state, and the state value returned is again a map associating each edge of G with the state value returned by the update_{op} invocation for that edge. □

Updating of a code graph can be optimized under the assumption that the individual update functions will return UNCHANGED results when all their inputs are UNCHANGED. That way, a complete traversal of the code graph can often be avoided.

E. Incremental Update of Error Checks and Loops

We can handle implicit loops and error handling by first using the translation given in Sections IV-C and IV-D to translate these features to explicit loop-boxes and if-boxes. We then treat the $(\texttt{LOOP}, G', f)$ and $(\texttt{IF}, G', f)$ families of edge labels as regular operations and define appropriate implementation functions for them.

The $\texttt{Opt}[t]$ types used for error handling are not aggregate datatypes. We do not need any special tag values beyond those defined for the underlying type t, so we define $\texttt{Tag}[\texttt{Opt}[t]] := \texttt{Tag}[t]$ for all types t.

To handle update for an if-box, the state value will be either the state for the contained graph, or the value ERROR, if the graph was not evaluated because the error check failed. This is used to decide whether the contents of the if box should be evaluated, updated or undone, and whether any ERROR outputs should be marked as NEW or as UNCHANGED.

The implementation of $\mathsf{update}_{(\texttt{IF},G',f)}$ can be seen in Listing 2; the implementations of $\mathsf{evaluate}_{(\texttt{IF},G',f)}$ and $\mathsf{undo}_{(\texttt{IF},G',f)}$ trivially forward to the corresponding functions on the contained graph G' and are therefore left out for brevity.

Dealing with repetition is more complicated, as $\texttt{List}[t]$ is an aggregate type. When individual items in a list are changed, we want to reevaluate only the corresponding iterations of loops that iterate over that list. The tag types $\texttt{Tag}[\texttt{List}[t]]$ therefore need to store individual tags for the list elements.

Definition 10: We define the tag for a list type to be either NEW, UNCHANGED or a list of tags for the individual list elements, or, more formally:

$$\texttt{Tag}[\texttt{List}[t]] := \{\texttt{NEW}, \texttt{UNCHANGED}\} \cup \texttt{Tag}[t]^*$$

where $*$ denotes the Kleene closure. □

Note that this definition does not allow tracking movement of elements within an array; the added complexity of such a system does not seem worthwhile at this time. Permuting or swapping list elements in response to a parameter change will therefore require all involved list elements to be marked as changed.

Listing 2 The update function for if boxes

```
function update_(IF,G',f)(s, arg_1...n, oldout_1...m)
    if all atag_i are UNCHANGED then
        return (s, oldout_1...m,
                   UNCHANGED ··· UNCHANGED)
    end if

    if any arg_i with f_i = 1 is ERROR then
        if s is ERROR then
            for all outputs do
                (out_i, otag_i) ← (ERROR, UNCHANGED)
            end for
        else
            undo_G'(s)
            for all outputs do
                (out_i, otag_i) ← (ERROR, NEW)
            end for
        end if
        s' ← ERROR
    else
        if s is ERROR then
            (s', out_1...m) ← evaluate_G'(arg_1...n)
            otag_1...m ← NEW
        else
            (s', out_1...m, otag_1...m)
                ← update_G'(s, arg_1...n, tag_1...n)
        end if
    end if
    return (s', out_1, otag_1 ··· out_m, otag_m)
end function
```

The persistent state s for a loop box is a list of the persistent states for the individual iterations. Thus, the update function will calculate the number of iterations required and compare it with the number of iterations done the previous time. States that are still needed are updated using $\mathsf{update}_{G'}$. If fewer iterations are needed, extra states are destroyed using $\mathsf{undo}_{G'}$. If the number of iterations has increased, new states are created using $\mathsf{evaluate}_{G'}$.

The update function for loop edges, $\mathsf{update}_{(\texttt{LOOP},G',f)}$, can be seen in Listing 3. Extracting the proper arguments for specific loop iterations happens as described in Section IV-C. The definitions of $\mathsf{evaluate}_{(\texttt{LOOP},G',f)}$ and $\mathsf{undo}_{(\texttt{LOOP},G',f)}$ are straightforward and are left out for brevity.

This concludes the extensions to the language (cf. Section IV). They cover the incremental updates of individual operations up to incremental updates of whole code graphs, as well as the handling of implicit loops and error checks in this context.

VII. Discussion and Conclusion

We have presented a formal framework for the representation of procedural models that is particularly suited for direct manipulation of procedural 3D content.

Listing 3 The update function for loop boxes

```
function update_(LOOP,G',f)(s, arg_1...n, oldout_1...m)
    if all atag_i are UNCHANGED then
        return (s, oldout_1...m,
                    UNCHANGED ··· UNCHANGED)
    end if

    n_new ← min(lengths of relevant input arrays)
    n_old ← length of s
    for i ← 1, max(n_new, n_old) do
        iarg_1...n ← extract arguments for iteration i
        itag_1...n ← extract tags for iteration i
        if i > n_old then
            (s'[i], out_1...m[i]) ← evaluate_G'(iarg_1...n)
            otag_1...m[i] ← NEW
        else if i > n_new then
            undo_G'(s[i])
        else
            (s'[i], out_1...m[i], otag_1...m[i])
                ← update_G'(s[i], iarg_1...n, itag_1...n)
        end if
    end for
    return (s', out_1...m, otag_1...m)
end function
```

The design space for the dataflow language is constrained by three main considerations: simplicity, efficiency and interactivity.

Simplicity in this case means minimizing the number of entities that do not have a natural visual representation in the GUI. A three-dimensional shape can be represented directly in the GUI; a repeated three-dimensional shape can also be represented. A repetition operator, on the other hand, is an abstract concept, not a three-dimensional object. By introducing *implicit* looping and error handling constructs (Section IV), we have avoided this problem.

Next is *efficiency*. Even simple procedural models can, by virtue of their procedural nature, generate relatively large amounts of geometry data; efficiency is thus always a concern. We have benchmarked the loop fusion and error handling optimizations on three different models. The code graphs are compiled to GML [2], a language syntactically similar to PostScript. The measurement is based on the number of executable statements, or tokens; this is independent of model parameters (repetition counts) and of the implementation quality of basic operations. See Table I for the results of optimizing loops (Opt A) and loops and error handling (Opt B).

TABLE I
OPTIMIZATION BENCHMARK: EFFECTS OF FUSING LOOPS (OPT A) AND LOOPS & ERROR HANDLING (OPT B) ON MODEL SIZE.

Model	Tokens	Opt A	Opt B
gothic ornament	1322	992	789
simple house	408	258	225
complex façade	69769	30846	24865

The third and final consideration was *interactivity*. Procedural models are not always evaluated from scratch; this is especially the case in an interactive procedural editor, where the user can adjust individual parameters of a model using the mouse. For a procedural modeling system to perform well in that situation, it needs to avoid doing unnecessary recalculations. We have found that it is not enough to do this at the level of individual objects, as the granularity of these objects is dictated by the requirement of simplicity. An entity perceived as a single object by the user might in fact consist of several parts that can be updated individually. The method we have described in Section VI addresses this by allowing the implementations of the modeling operations to cooperate in providing incremental update functionality.

Taken together, these individual aspects form a system that constitutes a solid base for a direct manipulation based graphical procedural modeler that can be used with different modeling vocabularies depending on the concrete application. Since the publication of our conference paper [1], we have successfully used systems based on this framework for different applications of procedural modeling [20], [21].

A. Future Work

Interactive performance could, in theory, be improved further by taking advantage of the fact that during interactive manipulation of a procedural model, the same parameters are often changed repeatedly. Applying a form of constant folding to the tag values described in Section VI might serve to eliminate a lot of redundant checking. Parallel execution of modeling operations would be very beneficial for large and complex models, as well.

There are also many research opportunities for adapting existing programming language concepts to our framework and to the context of direct manipulation procedural modeling. Defining modules or functions is a well-known technique, but it is unknown how well they can be adapted to the special requirements imposed by direct manipulation. Complex procedural 3D models will necessarily suffer from the same problems as complex software does in general; so at some point it will be necessary to investigate methods of 'shape refactoring'.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support from the Austrian Research Promotion Agency (FFG) for the project "V2me - Virtual Coach Reaches Out To Me" (825781), which is part of the AAL Joint Programme of the European Union.

REFERENCES

[1] W. Thaller, U. Krispel, S. Havemann, and D. Fellner, "Implicit nested repetition in dataflow for procedural modeling," in *COMPUTATION TOOLS 2012*, T. Ullrich and P. Lorenz, Eds. IARIA, 2012, pp. 45–50.

[2] S. Havemann, "Generative mesh modeling," Ph.D. dissertation, Technical University Braunschweig, 2005.

[3] Robert McNeel & Associates, "Grasshopper for Rhino3D," [retrieved: 2012, 05]. [Online]. Available: http://www.grasshopper3d.com/

[4] Side Effects Software, "Houdini," [retrieved: 2012, 05]. [Online]. Available: http://www.sidefx.com

[5] T. Green and M. Petre, "When visual programs are harder to read than textual programs," in *Proceedings of ECCE-6*, 1992, pp. 167–180.

[6] CGAL, "Computational Geometry Algorithms Library," [retrieved: 2012, 05]. [Online]. Available: http://www.cgal.org
[7] Adobe Inc., *PostScript Language Reference Manual*, 3rd ed. Addison-Wesley, 1999.
[8] Processing, "Processing," [retrieved: 2012, 05]. [Online]. Available: http://www.processing.org
[9] D. Gould, *Complete Maya programming: an extensive guide to MEL and the C++ API*, ser. Morgan Kaufmann series in computer graphics and geometric modeling. Morgan Kaufmann Publishers, 2003.
[10] Robert McNeel & Associates, "Rhinoceros 3D," [retrieved: 2012, 05]. [Online]. Available: http://www.rhino3d.com
[11] G. Stiny and J. Gips, "Shape grammars and the generative specification of painting and sculpture," in *The Best Computer Papers of 1971*. Auerbach, 1972, pp. 125–135.
[12] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant architecture," *Proc. SIGGRAPH 2003*, pp. 669 – 677, 2003.
[13] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. V. Gool, "Procedural modeling of buildings," in *ACM SIGGRAPH*, vol. 25, 2006, pp. 614 – 623.
[14] Esri, "CityEngine," [retrieved: 2012, 05]. [Online]. Available: http://www.esri.com/software/cityengine/
[15] W. M. Johnston, J. R. P. Hanna, and R. J. Millar, "Advances in dataflow programming languages," *ACM Comput. Surv.*, vol. 36, no. 1, pp. 1–34, Mar. 2004.
[16] G. Patow, "User-friendly graph editing for procedural buildings," *Computer Graphics and Applications, IEEE*, vol. PP, no. 99, p. 1, 2010.
[17] S. Barroso, G. Besuievsky, and G. Patow, "Visual copy & paste for procedurally modeled buildings by ruleset rewriting," *Computers & Graphics*, vol. 37, no. 4, pp. 238–246, 2013.
[18] D. Plump, "Term graph rewriting," in *Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools*, H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, Eds. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 1999, pp. 3–61.
[19] W. Kahl, C. Anand, and J. Carette, "Control-flow semantics for assembly-level data-flow graphs," in *Relational Methods in Computer Science*, ser. Lecture Notes in Computer Science, W. MacCaull, M. Winter, and I. Düntsch, Eds. Springer Berlin / Heidelberg, 2006, vol. 3929, pp. 147–160.
[20] R. Zmugg, U. Krispel, W. Thaller, S. Havemann, M. Pszeida, and D. W. Fellner, "A new approach for interactive procedural modelling in cultural heritage," in *Proc. Computer Applications & Quantitative Methods in Archaeology (CAA 2012)*, 2012.
[21] R. Zmugg, W. Thaller, M. Hecher, T. Schiffer, S. Havemann, and D. W. Fellner, "Authoring animated interactive 3D museum exhibits using a digital repository." in *VAST*, D. B. Arnold, J. Kaminski, F. Niccolucci, and A. Stork, Eds. Eurographics Association, 2012, pp. 73–80. [Online]. Available: http://dblp.uni-trier.de/db/conf/vast/vast2012.html#ZmuggTHSHF12
[22] Y. Baulac, "Un micromonde de géométrie, cabri-géométre," Ph.D. dissertation, Joseph Fourier University of Grenoble, 1990.
[23] W. Thaller, U. Krispel, R. Zmugg, S. Havemann, and D. W. Fellner, "Shape grammars on convex polyhedra," *Computers & Graphics*, vol. 37, no. 6, pp. 707 – 717, 2013, Shape Modeling International (SMI) Conference 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0097849313000861
[24] L. Krecklau and L. Kobbelt, "Procedural modeling of interconnected structures," *Computer Graphics Forum*, vol. 30, no. 2, pp. 335–344, 2011. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2011.01864.x
[25] D. Reiners, G. Voss, and C. Neumann, "OpenSG," 2013. [Online]. Available: http://www.opensg.org/

Data Minability Evaluation by Compression – An Experimental Study

Dan A. Simovici, Dan Pletea, and Saaid Baraty
University of Massachusetts Boston,
Boston, USA,
{dsim,dpletea,sbaraty}@ cs.umb.edu

***Abstract*—The effectiveness of compression algorithms is increasing as the data subjected to compression contains patterns which occur with a certain regularity. This basic idea is used to detect the existence of regularities in various types of data ranging from market basket data to undirected graphs. The results are quite independent of the particular algorithms used for compression and offer an indication of the potential of discovering patterns in data before the actual mining process takes place.**

Keywords-data mining; lossless compression; LZW; market basket data; patterns; Kronecker product.

I. INTRODUCTION

Our goal is to show that compression can be used as a tool to evaluate the potential of a data set of producing interesting results in a data mining process. The basic idea that data that contains patterns that occur with a certain regularity will be compressed more efficiently compared to data that has no such characteristics. Thus, a pre-processing phase of the mining process should allow to decide whether a data set is worth mining, or compare the interestingness of applying mining algorithms to several data sets.

Since compression is generally inexpensive (and certainly less expensive than mining algorithms), and compression methods are well-studied and understood, pre-mining using compression will help data mining analysts to focus their efforts on mining resources that can provide a highest payout without an exorbitant cost.

Compression has received lots of attention in the data mining literature. As observed by Mannila [14], data compression can be regarded as one of the fundamental approaches to data mining [14], since the goal of the data mining is to "compress data by finding some structure in it".

The role of compression developing parameter-free data mining algorithms in anomaly detection, classification and clustering was examined in [8]. The size $C(x)$ of a compressed file x is as an approximation of Kolmogorov complexity [4] and allows the definition of a pseudo-distance between two files x and y as

$$d(x,y) = \frac{C(xy)}{C(x)+C(y)},$$

where xy is the file obtained by concatenating x and y. Note that this is not the common definition of a pseudo-distance (see, for example [21]); instead, it is simply a numerical evaluation of the similarity of the files x and y; its minimal value is obviously equal to 0.5.

Further advances in this direction were developed in [9], [10] and [23]. A Kolmogorov complexity-based dissimilarity was successfully used to texture matching problems in [3] which have a broad spectrum of applications in areas like bioinformatics, natural languages. and music. Compression algorithms are used in the actual mining process to handle data mining explorations that return huge sets of results by extracting those results that actually are representative of the data set (see, for example [19], [22]).

Our goal in this paper is to show that compression can be used for assessing the interestingness of applying an actual data mining process. In other words, to evaluate the minability of a data set using compression. We justify experimentally this idea by evaluating data sets that have different characteristics and sources.

In general, data mining is task-oriented and the mining process entails seeking specific patterns. Thus, our assessment of minability will not necessarily help identify patterns of interest; instead, it will signal that such patterns may exist and will invite to further exploration.

There are two broad classes of compression algorithms: lossy compression, that reduces significantly data but does not allow the full inverse transformation, from compressed data to the original data, and lossless compression, that achieves data reduction and can be completely reversed. We illustrate the use of lossless compression in pre-mining data by focusing on several distinct data mining processes: files with frequent patterns, frequent item sets in market basket data, and exploring similarity of graphs.

The LZW (Lempel-Ziv-Welch) algorithm was introduced in 1984 by T. Welch in [24] and is among the most popular compression techniques. The algorithm does not need to check all the data before starting the compression and the performance is based on the number of the repetitions and the lengths of the strings and the ratio of 0s/1s or true/false at the bit level. There are several versions of the LZW algorithm. Popular programs (such as Winzip or the zip function of MATLAB) use variations of the LZW compression. These algorithms work both at the bit level and at the character level.

An important role in evaluating concentrations of values in various probability distributions is played by the notion of entropy, Namely, if $\mathbf{p} = (p_1, \ldots, p_n)$ is a probability distribution with $p_i \leqslant 0$ and $\sum_{i=1}^{n} p_i = 1$, the entropy of

this distribution is

$$\mathcal{H}(\mathbf{p}) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i}.$$

It is well-known (see [6], [13]) that the maximum value of the entropy is obtained when

$$p_1 = p_2 = \cdots = p_n = \frac{1}{n}$$

and this value is $\log_2 n$. The minium value is 0 and this occurs when there exists p_i with $p_i = 1$ and $p_j = 0$ for $j \neq i$, $1 \leqslant j \leqslant n$. The entropy helps us to evaluate the diversity of the values assumed by a random variable: the more concentrated these values are the lower the entropy.

After examining compressibility of binary strings in Section II we explore several experimental settings that provide strong empirical evidence of the correlation between compression ratio and the existence of hidden patterns in data. In Section III we discuss the compressibility of sequences of symbols produced by various generative mechanisms. Section IV is dedicated to the compressibility of adjacency matrix for graphs relative to the entropy of distribution of subgraphs. Finally, in Section V, we examine the compressibility of files that contain market basket data sets. This paper is an extension of our contribution [1].

II. Patterns in Strings and Compression

An *alphabet* is a finite and non-empty set whose elements are referred to as *symbols*. Let A^* be the set of sequences on the alphabet A. We refer to these sequences as *words* or *strings*. The length of a string w is denoted by $|w|$. The null string on A is denoted by λ and we define A^+ as $A^+ = A^* - \{\lambda\}$. The subsets of A^* are referred to as *languages* over A.

If $w \in A^*$ can be written as $w = utv$, where $u, v \in A^*$ and $t \in A^+$, we say that the pair (t, m) is an occurrence of t in w, where m is the length of u.

The occurrences (x, m) and (y, p) are overlapping if $p < m + |x|$ and $m < p + |y|$. If this is the case, $m < p$ and $p + |y| > m + |x|$ then there is a proper suffix of x that equals a proper prefix of y. If x is a word such that the sets of its proper prefixes and its proper suffixes are disjoint, there are no overlapping occurrences of x in any word.

The number of occurrences of a string x in a string w is denoted by $n_x(w)$. Clearly, we have $\sum\{n_a(w) \mid a \in A\} = |w|$ for any symbol $a \in A$. The *prevalence* of x in w is the number $f_x(w) = \frac{n_x(w) \cdot |x|}{|w|}$ which gives the ratio of the characters contained in the occurrences of t relative to the total number of characters in the string.

The result of applying a compression algorithm C to a string $w \in A^*$ is denoted by $C(w)$ and the *compression ratio* is the number

$$\mathsf{CR}_C(w) = \frac{|C(w)|}{|w|}.$$

We shall use the binary alphabet $B = \{0, 1\}$ and the LZW algorithm, the compression algorithm of the package `java.util.zip`, or the `zip` function of MATLAB.

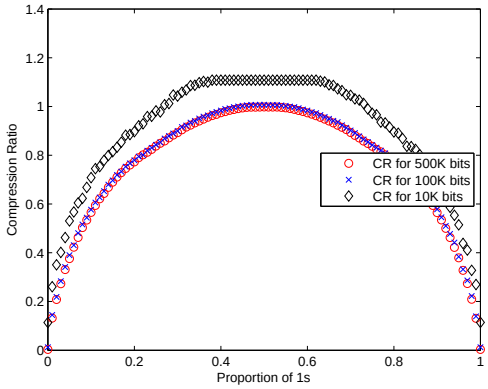


Figure 1. Baseline CR_{jZIP} Behavior

We generated random strings of bits (0s and 1s) and computed the compression ratio for strings with a variety of symbol distributions. A string w that contains only 0s (or only 1s) achieves a very good compression ratio of $\mathsf{CR}_{jZIP}(w) = 0.012$ for 100K bits and $\mathsf{CR}_{jZIP} = 0.003$ for 500K bits, where $jZIP$ denotes the compression algorithm from the package `java.util.zip`. Figure 1 shows, as expected, that the worst compression ratio is achieved when 0s and 1s occur with equal frequencies.

For strings of small length (less than 10^4 bits) the compression ratio may exceed 1 because of the overhead introduced by the algorithm. However, when the size of the random string exceeds 10^6 bits this phenomenon disappears and the compression ratio depends only on the prevalence of the bits and is relatively independent on the size of the file. Thus, in Figure 1, the curves that correspond to files of size $100K$ bits and $500K$ bits overlap. We refer to the compression ratio of a random string w that contains $n_0(w)$ zeros and $n_1(w)$ ones as the *baseline compression ratio* for this distribution of bits.

We created a series of binary strings $\varphi_{t,m}$ which have a minimum guaranteed number m of occurrences of patterns $t \in \{0, 1\}^k$, where $0 \leqslant m \leqslant 100$. The compression baselines for files containing the patterns 01, 001,0010, and 00010 are shown in Table I.

TABLE I. Baseline Compression Ratio for Files Containing a Minimum Guaranteed Number of Patterns

Pattern	Proportion of 1s	Baseline
01	50%	1.007
001	33%	0.934
0010	25%	0.844
00010	20%	0.779

Specifically, we created 101 files $\varphi_{001,m}$ for the pattern 001, each containing 100K bits and we generated similar series for $t \in \{01, 0010, 00010\}$. In the case of the 001 pattern the baseline is established at 0.934, and after the prevalence exceeds 20% the compression ratio drops dramatically. Results of the experiment for 001 are shown in Table II. In Figure 2 we show that similar results hold for all patterns mentioned

TABLE II. PATTERN '001' PREVALENCE VERSUS THE COMPRESSION RATIO CR_{jZIP}

Prevalence of '001' pattern	CR_{jZIP}
0%	0.93
10%	0.97
20%	0.96
30%	0.92
40%	0.86
50%	0.80
60%	0.72
70%	0.62
80%	0.48
90%	0.31
95%	0.19
100%	0.01

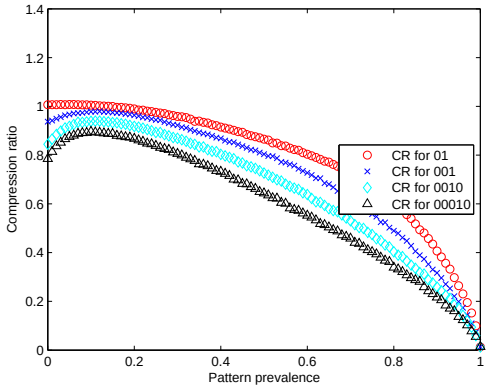


Figure 2. Dependency of Compression Ratio on Pattern Prevalence

above.

III. COMPRESSIBILITY OF LANGUAGES AND SEQUENCES

Sequences or sets of sequences of symbols are often subjected to data mining processes and identifying those sequences that contain interesting patterns before the actual mining process may be computationally significant.

We begin by examining the well-known sequence called the Thue-Morse sequence [2] that has many applications ranging from crystal physics [16], counter synchronization [25], metrology [7], [11], and chess playing [15], as well as in game theory, fractals and turtle graphics, chaotic dynamical systems, etc.

This sequence contains patterns but not repetitions.

Definition 3.1: Let $n \in \mathbb{N}$ be a natural number. The *Thue-Morse sequence* $\mathbf{s}_n = s_0 s_1 \cdots s_n$ is a word over the alphabet $\{0, 1\}$ defined as:

$$s_i = \begin{cases} 1 & \text{if } i \text{ has an odd number of 1s} \\ & \text{in its binary representation} \\ 0 & \text{otherwise,} \end{cases}$$

for $0 \leqslant i \leqslant n$. □

TABLE III. THE COMPRESSION RATIO $\mathsf{CR}_{jZIP}(\mathbf{s}_{2^k})$ FOR THUE-MORSE SEQUENCES

k	$\lvert seq_{2^k} \rvert$	$\mathsf{CR}_{jZIP}(seq_{2^k})$
5	32	34
8	256	4.625
10	1024	1.226
12	4096	0.328
14	16384	0.0932
15	32768	0.0542
16	65536	0.0322
17	131072	0.0208
18	262144	0.0151
19	524288	0.012
20	1048576	0.010
21	2097152	0.010
22	4194304	0.009

For example, we have

$$\mathbf{s}_{16} = (0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0).$$

It is clear that if $m, n \in \mathbb{N}$ and $m \leqslant n$, $\mathbf{s}_m$ is a prefix of $\mathbf{s}_n$. Thus, the successive Thue-Morse sequences define an infinite sequence.

An equivalent method for defining the Thue-Morse sequence is by starting with 0 and concatenating the complement of the sequence obtained so far. This procedure yields 0, then 01, 0110, 01101001, and so on. It is known (see [18], for example) that the Thue-Morse sequence is a cube-free sequence, that is, the sequence does not contain substrings of the form www.

We generated the Thue-Morse sequences and stored this sequence of 0s and 1s at the bit level. By using the zip compression utility from the `java.util.zip` package the compression ratios shown in Table III were obtained.

For small values of k, the sequence is incompressible due to the overhead produced by the compression process. As Table III and Figure 3 show, for k big enough ($2^k \geqslant 2000$) the sequence becomes compressible and the compression ratio reaches a low value (of less than 1%) for Thue-Morse sequences longer than $4,000,000$ characters. Since the Thue-Morse sequence $\mathbf{s}_{2^k}$ has equal number of 0s and 1s for any value of k and its compression ratio is well below the baseline compression ratio established for sequences of bits in Section II, we can conclude that even in the absence of repetitions, compression can be used for the detections of patterns.

In a series of experiments involving generative grammars we examined the compressibility of language fragments generated by these grammars. A generative grammar, or in short, a *grammar* is defined as a 4-tuple $G = (A_N, A_T, S, P)$, where A_N and A_T are non-empty, finite and disjoint sets referred to as the non-terminal and the terminal alphabet, respectively, $S \in A_N$ is the *initial symbol of the grammar* G, and P is a finite set of pairs of the form (α, β), where $\alpha \in (A_N \cup A_T)^+$ and $\beta \in (A_N \cup A_T)^*$. A pair $(\alpha, \beta) \in P$ is a production of the grammar G. Productions are used for rewriting words over $A_N \cup A_T$. Namely, if $\gamma, \delta \in (A_N \cup A_T)^*$, $\gamma = \gamma_1 \alpha \gamma_2$, and $\delta = \gamma_1 \beta \gamma_2$ for some production $(\alpha, \beta) \in P$, we write $\gamma \underset{G}{\Rightarrow} \delta$. The reflexive and transitive closure of the binary relation $\underset{G}{\Rightarrow}$ is denoted by "$\overset{*}{\underset{G}{\Rightarrow}}$". The language generated by G is the set

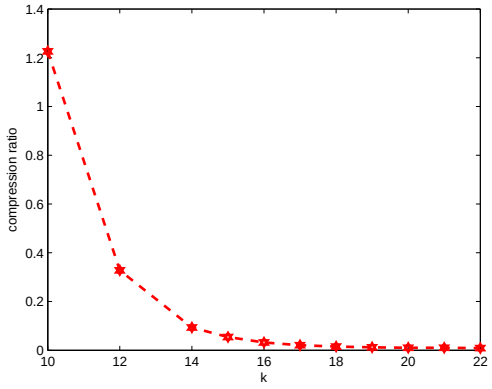


Figure 3. Compression Ratio Behavior of Thue-Morse Sequence

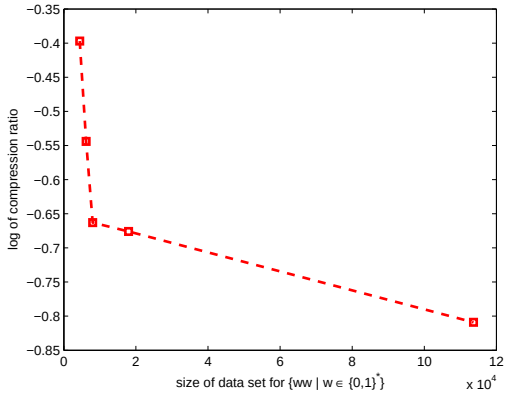


Figure 4. Compression Ratio Behavior of the language L_1

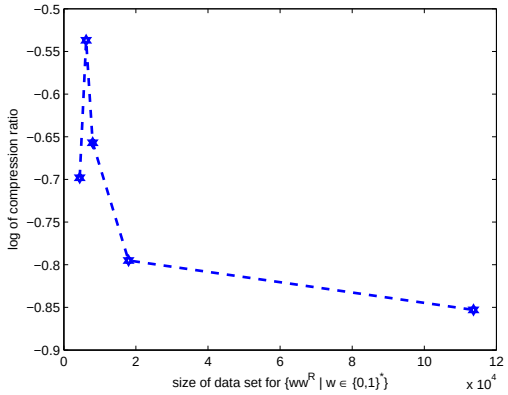


Figure 5. Compression Ratio Behavior of the language L_2

$L(G) = \{x \in A_T^* \mid S \stackrel{*}{\underset{G}{\Rightarrow}} \}$.

Grammars are used as generative devices that produce languages over their terminal alphabet. Chomsky's hierarchy (see [17] or [20]) defines four classes of grammars based on the complexity of their productions. In turn, these classes of grammars, define a strict hierarchy of classes of languages $\mathcal{L}_3 \subset \mathcal{L}_2 \subset \mathcal{L}_1 \subset \mathcal{L}_0$, where $\mathcal{L}_3$ is the class of regular languages, $\mathcal{L}_2$ is the class of context-free languages, $\mathcal{L}_1$ is the class of context-sensitive languages, and $\mathcal{L}_0$ is the class of recursively enumerable languages. It is worth noting that the classes $\mathcal{L}_3$ and $\mathcal{L}_2$ collapse on languages over one-symbol alphabet. In other words, if L is a language over an one-symbol alphabet, then $L \in \mathcal{L}_2$ implies $L \in \mathcal{L}_3$.

We evaluate the compressibility of a language L over an alphabet A by considering the increasing sequence of finite languages $\mathcal{S}(L) = (L_0, L_1, \ldots, L_n, \ldots)$, where L_n consists of the first n words of L in lexicographic order, computing the compression ratios $CR_{jZIP}(L_n)$, and examining the dependency of this ratio on n.

We examinine comparatively the compressibility of the languages $L_1 = \{ww \mid w \in \{0,1\}^*\}$ (a context-sensitive language) versus the compressibility of a similar language $L_2 = \{ww^R \mid w \in \{0,1\}^*\}$ (a context-free language) which has a simpler structure. Here, the word w^R is the *reversal* of the word w and is defined as $\lambda^R = \lambda$ and $(a_{i_1} \cdots a_{i_n})^R = a_{i_n} \cdots a_{i_1}$.

The results shown in Figures 4 and 5 show that L_2, the less complex language has a better (lower) compression ratio, and therefore, higher compressibility.

Similar results are obtained when comparing the compressibility of the context-sensitive languages L_{exp} and L_{prime} over the one-symbol alphabet $\{a\}$ defined by

$$\begin{aligned} L_{exp} &= \{a^{2^n} \mid n \in \mathbb{N}\}, \\ L_{prime} &= \{a^p \mid p \text{ is a prime number}\}. \end{aligned}$$

The reference [17] (see Chapter 1, section 2) contains specific grammars developed for both languages. Namely, the grammar for L_{exp} has 6 productions, while the second grammar that generates L_{prime} has 42 productions. As expected, experiments summarized in Figure 6 show that the L_{exp} is more compressible than L_{prime} which has a rather complex generating process.

These results suggest that the compressibility of languages is related to the complexity of the generative process that produce them. This will be the object of further investigations.

IV. Random Insertion and Compression

For a matrix $M \in \{0,1\}^{u \times v}$ denote by $n_i(M)$ the number of entries of M that equal i, where $i \in \{0,1\}$. Clearly, we have $n_0(M) + n_1(M) = uv$.

For a random variable $\mathcal{V}$ which ranges over the set of matrices $\{0,1\}^{u \times v}$ let $\nu_i(\mathcal{V})$ be the random variable whose values equal the number of entries of $\mathcal{V}$ that equal i, where $i \in \{0,1\}$.

Let $A \in \{0,1\}^{p \times q}$ be a $0/1$ matrix and let

$$\mathcal{B} : \begin{pmatrix} B_1 & B_2 & \cdots & B_k \\ p_1 & p_2 & \cdots & p_k \end{pmatrix},$$

be a matrix-valued random variable where $B_j \in \mathbb{R}^{r \times s}$, $p_j \geqslant 0$ for $1 \leqslant j \leqslant k$, and $\sum_{j=1}^{k} p_j = 1$.

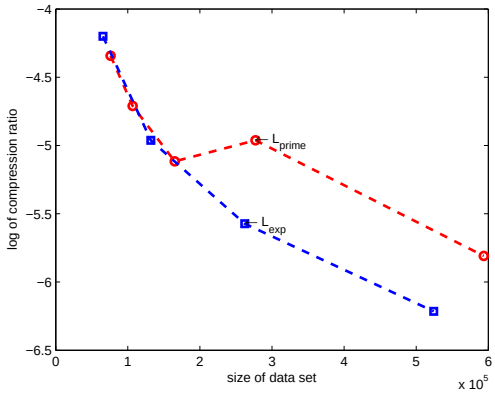


Figure 6. Compression Ratios of Languages L_{exp} and L_{prime}

Definition 4.1: The random variable $A \leftarrow \mathcal{B}$ obtained by the *insertion* of $\mathcal{B}$ into A is given by

$$A \otimes \mathcal{B} = \begin{pmatrix} a_{11}\mathcal{B} & \dots & a_{1n}\mathcal{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathcal{B} & \dots & a_{mn}\mathcal{B} \end{pmatrix} \in \mathbb{R}^{mr \times ns}$$

In other words, the entries of $A \leftarrow \mathcal{B}$ are obtained by substituting the block $a_{ij}B_\ell$ with the probability p_ℓ for a_{ij} in A. □

Note that this operation is a probabilistic generalization of Kronecker's product for if

$$\mathcal{B} : \begin{pmatrix} B_1 \\ 1 \end{pmatrix},$$

then $A \leftarrow B$ has as its unique value the Kronecker product $A \otimes B$.

The expected number of 1s in the insertion $A \leftarrow \mathcal{B}$ is

$$E[\nu_1(A \leftarrow \mathcal{B})] = n_1(A) \sum_{j=1}^{k} n_1(B_j) p_j$$

When $n_1(B_1) = \cdots = n_1(B_k) = n$, we have $E[\nu_1(A \leftarrow \mathcal{B})] = n_1(A)n$.

In the experiment that involves insertion, we used a matrix-valued random variable such that $n_1(B_1) = \cdots = n_1(B_k) = n$. Thus, the variability of the values of $A \leftarrow \mathcal{B}$ is caused by the variability of the matrices $B_1, \dots, B_k$ which can be evaluated using the entropy of the distribution of $\mathcal{B}$,

$$\mathcal{H}(\mathcal{B}) = -\sum_{j=1}^{k} p_j \log_2 p_j.$$

We expect to obtain a strong positive correlation between the entropy of $\mathcal{B}$ and the degree of compression achieved on the file that represents the matrix $A \leftarrow \mathcal{B}$, and the experiments support this expectation.

In a first series of experiments, we worked with a matrix $A \in \{0,1\}^{106 \times 106}$ and with a matrix-valued random variable

$$\mathcal{B} : \begin{pmatrix} B_1 & B_2 & B_3 \\ p_1 & p_2 & p_3 \end{pmatrix},$$

where $B_j \in \{0,1\}^{3\times 3}$, and $n_1(B_1) = n_1(B_2) = n_1(B_3) = 4$.

Several probability distributions were considered, as shown in Table IV. Values of $A \leftarrow \mathcal{B}$ had $106^2 * 3^2 = 101124$ entries.

In Table IV, we had 39% 1s and the baseline compression rate for a binary file with this ratio of 1s is 0.9775. We also computed the correlation between the CR_{jZIP} and the Shannon entropy of the probability distribution and obtained the value 0.98 for the insertion of a matrix-valued random variable having three values.

In Table V, we did the same experiment but with 4 different matrices of format 4×4. An even stronger correlation (0.99) was observed between CR_{jZIP} and the Shannon entropy of the probability distribution.

TABLE IV. INSERTION OF A THREE-VALUED RANDOM VARIABLE, ENTROPY AND COMPRESSION RATIOS

Probability distribution			Compression Ratio	Entropy
p_1	p_2	p_3		
0	1	0	0.33	0
1	0	0	0.33	0
0	0	1	0.33	0
0.9	0.1	0	0.51	0.46
0.8	0	0.2	0.61	0.72
0	0.3	0.7	0.7	0.88
0.2	0.2	0.6	0.77	1.37
0.6	0.2	0.2	0.74	1.37
0.15	0.35	0.5	0.78	1.44
0.49	0.25	0.26	0.77	1.5
0.33	0.33	0.34	0.79	1.58

TABLE V. INSERTION OF A FOUR-VALUED RANDOM VARIABLE, ENTROPY AND COMPRESSION RATIOS

Probability distribution				Compression Ratio	Entropy
p_1	p_2	p_3	p_4		
0	1	0	0	0.23	0
0.4	0	0.2	0.4	0.53	1.52
0.45	0.12	0.22	0.21	0.61	1.83
0.3	0.1	0.2	0.4	0.65	1.84
0.2	0.2	0.2	0.4	0.69	1.92
0.25	0.25	0.25	0.25	0.69	2

The relationship between the compression ratio CR_{jZIP} and the Shannon entropy of the probability distribution of the inserted random variable is shown in Figure 7 for both experiments.

This experiment reconfirms that data that contains patterns can be better compressed than randomly generated files and that the compressibility is less pronounced when the diversity of these patterns increases.

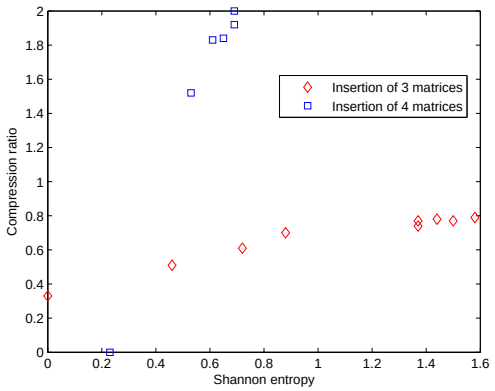


Figure 7. Evolution of CR_{jZIP} and Shannon Entropy for Insertions

Next, we examine the compressibility of binary square matrices and its relationship with the distribution of principal submatrices. An $m \times m$ *principal submatrix* of a matrix $A \in \mathbb{R}^{n\times n}$ is the matrix $A[I]$ defined by a non-empty m-element subset I of the set $\{1,\ldots,n\}$ and is obtained by selecting entries of A of the form a_{ij}, where $i,j \in I$. We mention that the principal submatrices of the adjacency matrix of a graph correspond to the adjacency matrices of the subgraphs of that graph. The patterns in a graph are captured in the form of frequent isomorphic subgraphs.

A binary square matrix is compressed by first vectorizing the matrix and then compressing the resulting binary sequence. There is a strong correlation between the compression ratio of the adjacency matrix of a graph and the frequencies of the occurrences of isomorphic subgraphs of it. Specifically, the lower the compression ratio is, the higher are the frequencies of isomorphic subgraphs and hence the worthier is the graph for being mined.

Let $\mathcal{G}_n$ be an undirected graph having $\{v_1,\ldots,v_n\}$ as its set of nodes. The adjacency matrix of $\mathcal{G}_n$, $\mathbf{A}_{\mathcal{G}_n} \in \{0,1\}^{n\times n}$ is defined as

$$(\mathbf{A}_{\mathcal{G}_n})_{ij} = \begin{cases} 1 & \text{if there is an edge between } v_i \text{ and } v_j \text{ in } \mathcal{G}_n \\ 0 & \text{otherwise.} \end{cases}$$

We denote with $\mathsf{CR}_C(\mathbf{A}_{\mathcal{G}_n})$ the compression ratio of the adjacency matrix of graph $\mathcal{G}_n$ obtained by applying the compression algorithm C.

Let $S = \{i_1,\ldots,i_k\}$ be a subset of $\{1,\ldots,n\}$. The principal submatrix $\mathbf{A}_{\mathcal{G}_n}[S]$ is the adjacency matrix of the subgraph of $\mathcal{G}_n$ which consists of the nodes with indices in S along with those edges that connect these nodes. We denote by $\mathcal{P}_n(k)$ the collection of all subsets of $\{1,2,\ldots,n\}$ of size k where $2 \leqslant k \leqslant n$. We have $|\mathcal{P}_n(k)| = \binom{n}{k}$.

Let $(\mathbf{A}_1^k,\ldots,\mathbf{A}_{\ell_k}^k)$ be an enumeration of possible adjacency matrices of graphs with k nodes where $\ell_k = 2^{\frac{k(k-1)}{2}}$. We define the finite probability distribution

$$P(\mathcal{G}_n,k) = \left(\frac{\mathsf{n}_1^k(\mathcal{G}_n)}{|\mathcal{P}_n(k)|},\ldots,\frac{\mathsf{n}_{\ell_k}^k(\mathcal{G}_n)}{|\mathcal{P}_n(k)|}\right),$$

where $\mathsf{n}_i^k(\mathcal{G}_n)$ for $1 \leqslant i \leqslant \ell_k$ is the number of subgraphs of $\mathcal{G}_n$ with adjacency matrix $\mathbf{A}_i^k$. The Shannon entropy of this probability distribution is:

$$\mathcal{H}_P(\mathcal{G}_n,k) = -\sum_{i=1}^{\ell_k} \frac{\mathsf{n}_i^k(\mathcal{G}_n)}{|\mathcal{P}_n(k)|} \log_2 \frac{\mathsf{n}_i^k(\mathcal{G}_n)}{|\mathcal{P}_n(k)|}.$$

If $\mathcal{H}_P(\mathcal{G}_n,k)$ is low, there are to be fewer and larger sets of isomorphic subgraphs of $\mathcal{G}_n$ of size k. In other words, small values of $\mathcal{H}_P(\mathcal{G}_n,k)$ for various values of k suggest that the graph $\mathcal{G}_n$ contains repeated patterns and is susceptible to produce interesting results. Note that although two isomorphic subgraphs do not necessarily have the same adjacency matrix, the number $\mathcal{H}_P(\mathcal{G}_n,k)$ is a good indicator of the diversity of isomorphic subgraphs and hence of the frequency subgraph patterns.

We evaluated the correlation between $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ and $\mathcal{H}_P(\mathcal{G}_n,k)$ for different values of k.

As expected, the compression ratio of the adjacency matrix and the distribution entropy of graphs are roughly the same for isomorphic graphs, so both numbers are characteristic for an isomorphism type. If ϕ is a permutation of the vertices of $\mathcal{G}_n$, the adjacency matrix of the graph $\mathcal{G}_n^\phi$ obtained by applying the permutation is defined by $\mathbf{A}_{\mathcal{G}_n^\phi}$ is given by

$$\mathbf{A}_{\mathcal{G}_n^\phi} = P_\phi \mathbf{A}_{\mathcal{G}_n} P_\phi^{-1}.$$

We compute the adjacency matrix $\mathbf{A}_{\mathcal{G}_n^\phi}$, the entropy $\mathcal{H}_P(\mathcal{G}_n^\phi,k)$, and the compression ratio $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n^\phi})$ for several values of k and permutations.

Graphs with $n = 60$ nodes and various number of edges ranging from 5 to 1765 were randomly generated. For each generated graph, we randomly produced twenty permutations of its set of nodes and computed $\mathcal{H}_P(\mathcal{G}_n^\phi,k)$ and $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n^\phi})$.

Finally, for each graph we calculated the ratio of standard deviation over average for the computed compression ratios, followed by the same computation for distribution entropies.

The results of this experiment are shown in Figures 8 and 9 against the number of edges. As it can be seen, the deviation over mean of the compression ratios for $n = 60$ does not exceed the number 0.05. Also, the deviation over average of the distribution entropies for various values of k do not exceed 0.006. In particular, the deviation of the distribution entropy for the graphs of 100 to 1500 edges falls below 0.001, which allows us to conclude that the deviations of both compression ratio and distribution entropy with respect to isomorphisms are negligible.

For each $k \in \{3,4,5\}$, we generated randomly 560 graphs having 60 vertices and sets of edges whose size were varying from 10 to 1760. Then, the numbers $\mathcal{H}_P(\mathcal{G}_n,k)$ and $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ were computed. Figure 10 captures the results of the experiment. Each plot contains two curves. The first curve represents the changes in average $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ for forty randomly generated graphs of equal number of edges. The second curve represents the variation of the average $\mathcal{H}_P(\mathcal{G}_n,k)$ for the same forty graphs. The trends of these two curves are very similar for different values of k.

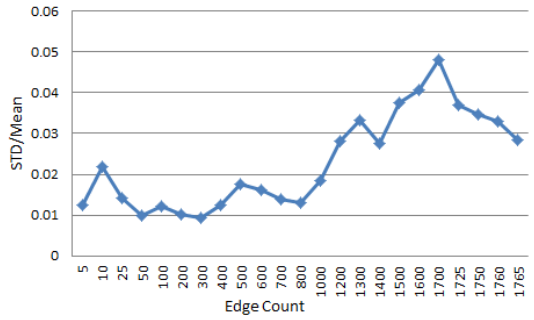


Figure 8. Standard deviation vs. average of the $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ for a number of different permutations of nodes for the same graph. The horizontal axis is labelled with the number of edges of the graph.

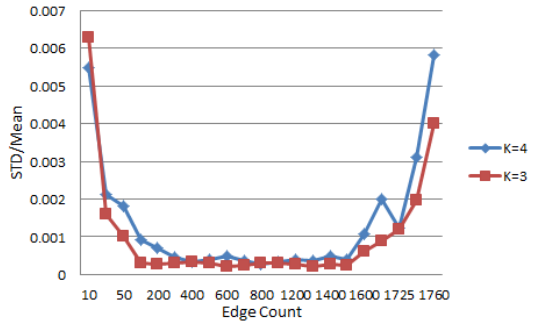


Figure 9. Standard deviation vs. average of the $\mathcal{H}_P(\mathcal{G}_n, k)$ of a number of different permutations of nodes for the same graph. The horizontal axis is labelled with the number of edges of the graph. Each curve corresponds to one value of k.

Table VI contains the correlation between $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ and $\mathcal{H}_P(\mathcal{G}_n, k)$ calculated for the 560 randomly generated graphs for each value of k.

TABLE VI. CORRELATIONS BETWEEN $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ AND $\mathcal{H}_P(\mathcal{G}_n, k)$

k	Correlation
3	0.92073175
4	0.920952812
5	0.919256573

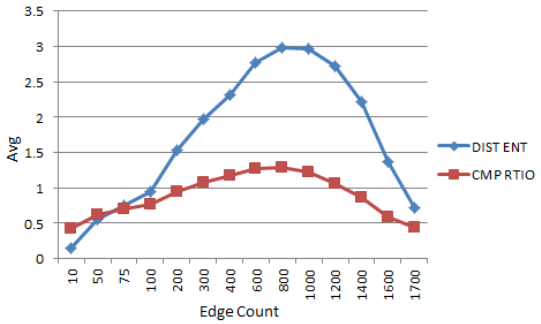
V. FREQUENT ITEMS SETS AND COMPRESSION RATIO

A market basket data set consists of a multiset $\mathcal{T}$ of *transactions*. Each transaction T is a subset of a set of items $I = \{i_1, \ldots, i_N\}$. The multiplicity of a transaction T in the multiset $\mathcal{T}$ is denoted by $m(T)$.

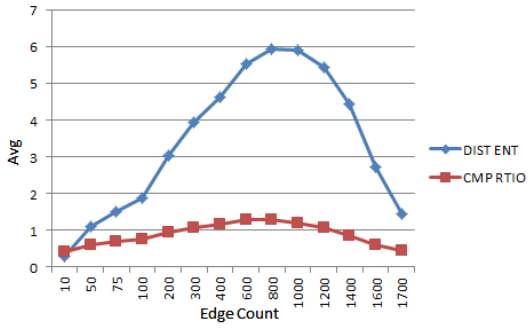
A transaction is described by its characteristic N-tuple $t = (t_1, \ldots, t_N)$, where

$$t_k = \begin{cases} 1 & \text{if } i_k \in T. \\ 0 & \text{otherwise,} \end{cases}$$

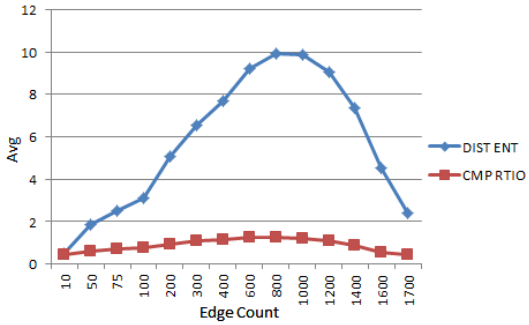
for $1 \leqslant k \leqslant N$. The length of a transaction T is $|T| = \sum_{k=1}^{N} t_k$, while the average size of transactions is $\frac{\sum\{|T| \mid T \text{ in } \mathcal{T}\}}{|\mathcal{T}|}$.



$n = 60$ and $k = 3$



$n = 60$ and $k = 4$



$n = 60$ and $k = 5$

Figure 10. Plots of average $\mathsf{CR}_{jZIP}(\mathbf{A}_{\mathcal{G}_n})$ (CMP RTIO) and average $\mathcal{H}_P(\mathcal{G}_n, k)$ (DIST ENT) for randomly generated graphs $\mathcal{G}_n$ of equal number of edges with respect to the number of edges.

The support of a set of items K of the data set $\mathcal{T}$ is the number

$$\mathsf{supp}(K) = \frac{|\{T \in \mathcal{T} \mid K \subseteq T\}|}{|\mathcal{T}|}.$$

The set of items K is s-frequent if $\mathsf{supp}(K) > s$.

The study of market basket data sets is concerned with the identification of association rules. A pair of item sets (X, Y) is an association rule, denoted by $X \to Y$. Its support, $\mathsf{supp}(X \to Y)$ equals $\mathsf{supp}(X)$ and its confidence $\mathsf{conf}(X \to Y)$ is defined as

$$\mathsf{conf}(X \to Y) = \frac{\mathsf{supp}(X \cup Y)}{\mathsf{supp}(X)}.$$

Using the artificial transaction ARMiner generator described in [5], we created a basket data set.

Transactions are represented by sequences of bits $(t_1, \cdots, t_N)$. The multiset $\mathcal{T}$ of M transactions was represented as a binary string of length MN obtained by concatenating the strings that represent transactions.

We generated files with 1000 transactions, with 100 items available in the basket, adding up to 100K bits.

For data sets having the same number of items and transactions, the efficiency of the compression increases when the number of patterns is lower (causing more repetitions). In an experiment with an average size of a frequent item set equal to 10, the average size of a transaction equal to 15, and the number of frequent item sets varying in the set $\{5, 10, 20, 30, 50, 75, 100, 200, 500, 1000\}$, the compression ratio had a significant variation ranging between 0.20 and 0.75, as shown in Table VII. The correlation between the number of patterns and the compression ratio was 0.544. Although the frequency of 1s and baseline compression ratio were roughly constant (at 0.75), the number of patterns and compression ratio were correlated.

TABLE VII. NUMBER OF ASSOCIATION RULES AT 0.05 SUPPORT LEVEL AND 0.9 CONFIDENCE

Number of Patterns	Frequency of 1s	Baseline compression	Compr. ratio	Number of rules
5	16%	0.75	0.20	9,128,841
10	17%	0.73	0.34	4,539,650
20	17%	0.73	0.52	2,233,049
30	17%	0.76	0.58	106,378
50	19%	0.75	0.65	2,910,071
75	18%	0.75	0.67	289,987
100	18%	0.75	0.67	378,455
200	18%	0.75	0.70	163
500	18%	0.75	0.735	51
1000	18%	0.75	0.75	3

Further, there was a strong negative correlation (-0.92) between the compression ratio and the number of association rules indicating that market basket data sets that satisfy many association rules are very compressible.

VI. CONCLUDING REMARKS

Compression ratio of a file can be computed fast and easy, and in many cases offers a cheap way of predicting the existence of embedded patterns in data. Thus, it becomes possible to obtain an approximative estimation of the usefulness of an in-depth exploration of a data set using more sophisticated and expensive algorithms.

The presence of patterns in strings leads to a high degree of compression (that is, to low compression ratios). Thus, a low compression ratio for a file indicates that the mining process may produce interesting results. Compressibility however, does not guarantee that a sequence contains repetitions. Strings that are free of repetitions but contain patterns can display a high degree of compressibility as shown by the well-known Thue-Morse binary string.

The use of compression as a measure of minability is illustrated on a variety of paradigms: graph data, market basket data, etc. Compression has been applied in bioinformatics as a tool for reducing the size of immense data sets that are generated in the genomic studies. Furthermore, specialized algorithms were developed that mine data in compressed form [12].

Our current work shows that identifying compressible areas of human DNA by comparing the compressibility of certain genomic regions is a useful tool for detecting areas where the gene replication mechanisms are disturbed (a phenomenon that occurs in certain genetically based diseases).

AKNOWLEDGEMENTS

This work was supported by a Science and Technology grant from the President of the University of Massachusetts.

The authors appreciate the remarks of the anonymous reviewers who contributed to the improvement of this paper.

REFERENCES

[1] D. Simovici, D. Pletea, and S. Baraty. Evaluating data minability through compression - an experimental study. In *Proceedings of DATA ANALYTICS*, pages 97–102. Think Mind Digital Library, 2012.

[2] J.-P. Allouche and J. Shallit. The ubiquitous Prouhet-Thue-Morse sequence. In *Sequences and their Applications*, pages 1–16. Springer London, 1999.

[3] B. J. L. Campana and E. J. Keogh. A compression based distance measure for texture. In *SDM*, pages 850–861, 2010.

[4] R. Cilibrasi and P. M. B. Vitànyi. Clustering by compression. *IEEE Transactions on Information Theory*, 51:1523–1545, 2005.

[5] L. Cristofor. The ARMiner project, 2000, Univ. of Massachusetts Boston. http://www.cs.umb.edu/ ∼laur/ARMiner.

[6] I. Csiszár and J. Körner. Information Theory - Coding Theorems for Discrete Memoryless Systems. (second edition). Cambridge University Press, Cambridge, UK, 2011.

[7] L. Jin, K. L. Parthasarathy, T. Kuyel, R. L. Geiger, and D. Chen. High-performance adc linearity test using low-precision signals in non-stationary environments. In *Proceedings 2005 IEEE International Test Conference*, pages 1182–1191, 2005.

[8] E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *Proc. 10th ACM SIGKDD Intnl Conf. Knowledge Discovery and Data Mining*, pages 206–215. ACM Press, 2004.

[9] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S. Lee, and J. Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14:99–129, 2007.

[10] E. J. Keogh, L. Keogh, and J. Handley. Compression-based data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 278–285. 2009.

[11] T. Kuyel D. Chen L. Jin, K. Parthasarathy and R. Geiger. Accurate testing of analog-to-digital converters using low linearity signals with stimulus error identification and removal. *IEEE Transactions on Instrumentation and Measurement*, 54:1188–1199, 2005.

[12] P. R. Loh, M. Baym, and B. Berger. Compressive genomics. *Nature Biotechnology*, 30:627–630, 2012.

[13] D. J. C. McKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge, UK, 2004.

[14] H. Mannila. Theoretical frameworks for data mining. *SIGKDD Exploration*, 1:30–32, 2000.

[15] M. Morse and G. A. Hedlund. Unending chess, symbolic dynamics, and a problem in semigroups. *Duke Mathematical Journal*, 11:1–7, 1944.

[16] L. Youran R. Ricklund, S. Mattias. The Thue-Morse aperiodic crystal, a link between the Fibonacci quasicrystal and the periodic crystal. *International Journal of Modern Physics B*, 1:121–132, 1987.

[17] A. Salomaa. *Formal Languages*. Academic Press, New York, 1973.

[18] A. Salomaa. *Jewels of Formal Language Theory*. Computer Science Press, Rockville, Maryland, 1981.

[19] A. Siebes, J. Vreeken, and M. van Leeuwen. Items sets that compress. In *Proceedings of SIAM International Conference on Data Mining*, pages 393–404. SIAM, 2006.

[20] D. A. Simovici and R. L. Tenney. *Formal Language Theory with Applications*. World Scientific, Singapore, 1999.

[21] D. A. Simovici and C. Djeraba. *Mathematical Tools for Data Mining*. (second edition) Springer, London, 2014.

[22] J. Vreeken, M. van Leeuwen, and A. Siebes. KRIMP: mining items that compress. *Data Mining and Knowledge Discovery*, 23:169–214, 2011.

[23] L. Wei, J. Handley, N. Martin, T. Sun, and E. J. Keogh. Clustering workflow requirements using compression dissimilarity measure. In *ICDM Workshops*, pages 50–54, 2006.

[24] T. Welch. A technique for high performance data compression. *IEEE Computer*, 17:8–19, 1984.

[25] R. Yarlagadda and J. Hershey. Counter synchronization using the Thue-Morse sequence and psk. *IEEE Transactions on Communications*, 32:947–977, 1984.

Complex Event Processing for Decision Support in an Airport Environment

Gabriel Pestana[1], Sebastian Heuchler[3], Augusto Casaca[1], Pedro Reis[2], and Joachim Metter[3]

[1]INESC-ID/INOV/IST, Lisboa, Portugal
gabriel.pestana, augusto.casaca@inesc-id.pt

[2]ANA-Aeroportos de Portugal, Lisboa, Portugal
pereis@ana.pt

[3]BIJO-DATA GmbH, Heldburg, Germany
sheuchler, jmetter@bijodata.de

***Abstract*— A new approach is proposed to the surveillance of Security and Safety occurrences concerning mobile objects in an airport environment, in particular to monitor aircrafts, vehicles and staff at the manoeuvring area for all weather conditions. A middleware platform receives localization information from the different mobile objects in the airport and merges that information through data fusion in the platform. The system outputs are shown in a high-resolution Graphical-User interface, providing a collaborative environment with the relevant information to the airport stakeholders. The outputs can be used by the stakeholders to take decisions on the best way to improve security and safety and also on the optimization of airport operational procedures in compliance with existing business rules. In this paper, the proposed system architecture follows an event-driven approach based on streams of occurrences processed in real-time. Therefore, it is suited for decision support. We will illustrate our approach by monitoring events occurred in an airport environment.**

Keywords-Mobility management; Situation awareness; Safety and security business rules; Location based services.

I. INTRODUCTION

In the airport environment, about 90% of critical events relate to accidents and incidents during ground handling services assisting parked aircrafts. The need for coordination of multiple activities occurring simultaneously requires, therefore, a continuous control of all ground movements, in particular during taxi operations. However, the current lack of context awareness and controllability is frequently identified as a causal factor for business rule infringements.

Without a solution capable of providing, in real-time, information related to the surveillance of operational occurrences, airport stakeholders (e.g., Airport Authority, Ground Handlers, Airlines) have not a reliable view of the overall situation to take well informed, in-time decisions [1] [2]. To assist airport stakeholders in their daily decision-making process we need an event-driven solution to combine data from multiple sources, capable of identifying meaningful events and responding to them as quickly as possible [3]. For instance, automatic detection of events related to over-speeding, safety infringements, unauthorized movements in restricted access areas, or any other location-based occurrence related to airport resources, staff or passengers.

Such capabilities would provide airport stakeholders with a new way to detect and analyse events in real-time, enabling them to adjust control actions according to the severity level of the observed event. However, to reach this level of operational intelligence and to avoid stakeholders to be drowned in data and be left without actionable information, they need to be assisted with data integration and fusion capability of unrelated events, based on business rules and policies. In this paper, the proposed system architecture follows an event-driven approach based on streams of occurrences in an airport environment. In particular, we will reference the running SECAIR project, whose platform will be deployed at Faro airport, Portugal.

The SECAIR project [4], is an European R&D project partially funded in the Eurostars program, brings a new approach to the surveillance of ground movements at the manoeuvring areas in the airport. It combines different localization technologies to detect and analyse movement patterns inside the airport terminal and at the apron area. The project relies on the development of an event observer system, which is capable of automatically identifying events and generating alarms in real-time. This means that a sliding window of one second is used to continuously provide streaming data to update the position of each surveyed object. A middleware platform provides data fusion to determine the localization of objects, which is determined by radio tracking techniques and video technology.

The middleware is part of a larger platform that, on the whole, will manage events related to movement patterns or hazardous situations. And because the project operates with different localization technologies simultaneously, multiple objects are surveyed, causing a very high volume of fine-grained data, which must be processed to determine movement patterns.

Current software architectures of decision support systems cannot deal efficiently with the processing of continu-

ous event streams. Existing approaches focus on knowledge processing, but do not explicitly target the problems associated to real-time event processing. [5].

To test the capabilities of the system, a set of business scenarios addressing airport operational requirements were defined in close collaboration with ANA-Aeroportos de Portugal - the main Portuguese airport's management company, based on the following needs:

- Traceability of vehicles and Ground Support Equipment with automatic detection of unauthorised incursions into restricted access areas;
- Tracking and controlling of ground handling operations;
- Surveillance of aircraft ground movements within the apron area;
- Provision of context awareness about on-going operations at the apron area, triggering safety and security alerts with different levels of severity;
- Support the decision making process by providing a reliable view of the overall situation whenever a safety or security event is reported;
- Ensure that event notifications are sent to airport stakeholders based on their roles/operational needs.

The paper is organized as follows: Section II introduces the process followed for the specification of the system requirements. Section III presents the main software components within the multi-tier architecture designed for the SECAIR system. Section IV presents the system implementation. Section V describes the use case of the project together with the operational scenarios defined for testing the system. Section VI reports on related work. Finally, conclusions are included in Section VII.

II. System Requirements Specification Process

An airport, which is usually classified as a critical infrastructure, needs to encompass functionalities for an unambiguous surveillance of surface traffic caused by aircrafts and vehicles, without reducing the safety level [6].

In the airport environment, to efficiently coordinate ground movements caused by aircrafts, passengers and cargo, decision makers must be able to respond to an increasingly complex range of events. To reach such level of continuous data integration for decision-making, airport stakeholders must have a graphical informational cockpit capable of communicating large amounts of relevant information in an intuitive way. This is a feature typically assigned to corporate spatial dashboards.

Indeed, when combining human understanding with data visualization techniques it is possible to track the situation awareness and simultaneously get a well picture of the business operational performance. This is particularly true when the visualization of key performance indicators (KPI), describing how business is performing, are correlated with the representation of on-going events as point features over a cartographic layout. To cope with such goals, core operational requirements were specified in close collaboration with airport stakeholders, enabling us to express the requirements in terms of features that the SECAIR system should satisfy.

Within the SECAIR project the collaboration of airport stakeholders during the requirements specification phase followed the recommendations of the IEEE 42010 standard. [7]. The standard states that a well-formed requirement is a statement of the system functionality that must be met or possessed by the system to solve stakeholder's objectives, and that is qualified by measurable conditions and bounded by constraints. In such approach, user requirements also generate a structured collection of information that embodies the requirements of the system mapped to the stakeholders concerns.

Figure 1 presents the schema of the system requirements specification (SyRS) process adopted within the SECAIR project for capturing safety and security requirements. The approach provided a "black box" description of what the system should do, in terms of the system's interactions or interfaces with its external environment. Therefore, the SyRS was essentially used as a technique to discover and document, through elicitation sessions with airport stakeholders, the system capabilities and its required behaviour. Besides distinguishing between requirements and their attributes (conditions and constraints), the SyRS also helped identifying, for each operational scenario, the corresponding business indicators used to validate whether the results (i.e., movement path of a surveyed object) are within acceptable thresholds.

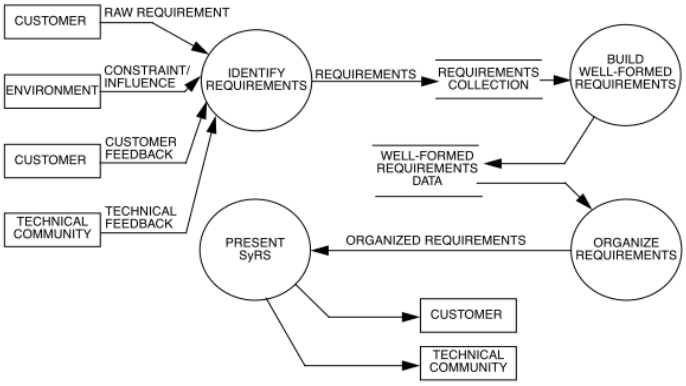


Fig. 1. System Requirements Specification process.

Afterwards, technical requirements, expressed as constraints placed on the SECAIR system, were analysed having in view the technical limits of each underlying technology. Besides these technical requirements, environmental influences affecting the system were also considered and classified into categories (e.g., political, standards and organizational policies). SECAIR conducted a research in each of these categories to ensure that the system conforms to all regulations that influence the airport sector.

The combination of all these artefacts was used to shape the set of operational scenarios that hold a high potential to demonstrate the benefits of the system developed under the SECAIR project.

For instance, in order to optimize gate to gate operations, the important issue to be considered is related with aircraft assisting tasks and the surveillance of ground movements in order to manoeuvre safely and efficiently on the movement area.

Concerning the objectives of SECAIR, in this paper, we are just considering the apron area and the service roads in the airport. The apron is a defined area intended to accommodate aircraft for purposes of loading and unloading passengers, mail and cargo, fuelling and parking. Its design should take into account safety procedures for aircraft manoeuvring, taking into account the specified clearances and following the established procedures to enter, move within and depart from apron areas.

By analyzing the activities and interactions that occur between aircraft and ground vehicles, we conclude that most of them take place in the apron areas. It is a wide variety of complex operations, including the handling of aviation fuel, the movements of vehicles, aircrafts and airport staff with different tasks to perform. They are all concentrated in a restricted area and with a short turnaround time, increasing the possibility of a potential conflict. Most of the time the activities performed in the apron are of vital importance for the safety of an aircraft during its subsequent flight.

Regarding the aircraft servicing mentioned above, we can refer to the following aircraft servicing operations involving ground vehicles and equipments:

- Passenger, baggage and cargo loading/unloading;
- Galley service;
- Fuelling service;
- Provision of compressed air for engine starting;
- Aircraft maintenance;
- In some cases, electric power and air conditioning.

As illustrated in Figure 2, there is a wide variety of ground operations, which contribute exponentially to conflicts and increase the risk of accidents/incidents in the apron area. Figure 2 outlines some of the vehicles used to assist parked aircrafts. The coordination of the movements of all those vehicles, sometimes with aircrafts parked at adjacent stand areas requires an effort to avoid delays and to comply with safety procedures. Besides being a very constrained area, in extreme situations (e.g., rush hours, bad meteorological conditions) the risk of operational inefficiencies at the stand area can compromise airport procedures, leading to a dysfunction of the airport and, eventually, compromising the required level of safety. Such stressing situation tends to increase the need to accurately monitor all ground movements within different areas in the airport air side, namely inside the stand area. However, without a system capable to continuously and accurately track all ground movements (i.e., vehicles, equipment and persons) inside restricted access areas, infringements to safety rules might not be noticed by apron controllers.

In the apron area, the most common type of incidents and accidents fall into the following categories:

- Ground equipment driven into aircraft
- Unmanned equipment rolls into aircraft
- Aircraft rolls forward/backward
- Towing vehicle strikes aircraft
- Aircraft contacts object/equipment.

Concerning the service roads, every effort should be made to plan air side service roads so that they do not cross runways and taxiways. Several solutions can be found to minimize the possibility of conflicts between aircrafts and vehicles/equipments, and one of them is considering road tunnels avoiding the crossings at taxiways. However, in all situations, vehicle drivers must comply with aerodrome regulations and take due care and attention to avoid collisions between vehicles and aircraft and other related hazards.

The SECAIR project addresses these concerns by monitoring all ground movements and by triggering alert messages for each detected infringement. The localization of each moving object is represented as point features, labelled with a colour code to call the attention of the controller (at the situation room) whenever a safety infringement is detected.

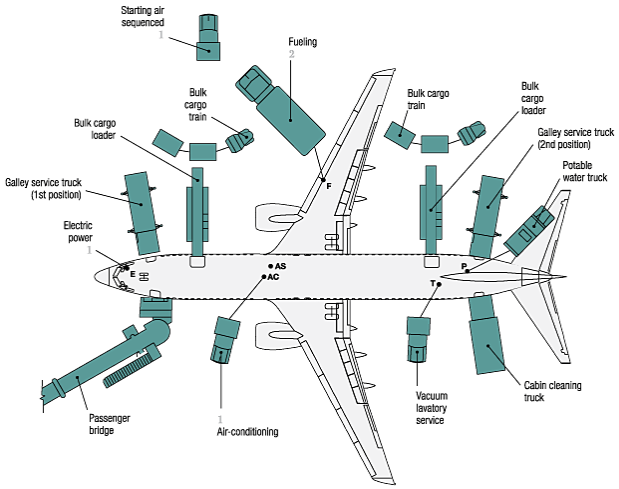


Fig. 2. The B-777 being serviced during a turnaround with the help of ground systems and mobile equipment. Source: Boing 777 Airplane Characteristics for Airport Planning.

Integrated airport operations planning, advanced surveillance techniques, ground-based safety nets and new runway management tools are amongst the improvements that will allow the aircraft to be served more efficiently from gate to gate. The SECAIR project will contribute to reduce ground hazards that affect flight safety, reduce aircraft ground damage, reduce personnel injuries and also, in the security domain, prevent acts of unlawful interference.

III. SYSTEM ARCHITECTURE

An event-driven solution typically consists of event observers (i.e., localization technologies) and event consumers. The SECAIR system operates by observing a set of events that happen in the external environment. Because localization technologies are continuously emitting data, they are particularly suited for Complex Event Processing (CEP). As outlined in Figure 3, an event starts at the Communication

tier, with the sensing of a fact (e.g., safety occurrence) that is converted into a data stream and sent to the Data Fusion Algorithm (DFA) at the middleware of the Application tier. To this end, any update to the position of each surveyed object is provided as a single time streaming data.

Location-based data tend to be strongly correlated in both time and space. For instance, position and speed data measured by one localization technology is highly correlated to the data collected by another adjacent localization technology. Similarly, readings observed at one time instant are highly indicative of the readings observed at the next time instant. This is particularly relevant because airport stakeholders are not interested in individual readings in time or individual devices in space, but rather in application-level concepts of temporal and spatial granularities.

In this paper, we use the term spatio-temporal to designate data related to both time and space dimensions. Since the project is closely related with location-based data, the trigger to most events derives from the movement of the observed objects. For each reported position, the system might require access to additional data about the object in order to analyse the event in conformity with the role of the object or to consider changes in the object status, changes in the object descriptive data or even changes in the spatial context where the object is located. The complexity to support such in-time actions increases when the system has to process data from multiple technologies while considering a set of business rules to take appropriate actions.

A. Overview

The generic architecture of the SECAIR system is shown in Figure 3. At the periphery of the system we have the Event Observers and the Event Consumers. The former corresponds to the devices and localization technologies, whereas the latter receives event notifications and presents them to the end-users so that they can react accordingly.

The SECAIR system implements a client-server architecture structured into three tiers (see Figure 4). The Communication tier operates with heterogeneous wireless localization technologies (sensors), each one collecting data about the location of the observed objects. Each device of the adopted localization technologies is responsible to continuously generate location-based data streams; for some technologies the period is less than one second. However, the sensor data is too fine-grained and do not meet airport stakeholder concerns as they are not interested in individual sensors data, but in application-level concepts of spatio-temporal granularities. Therefore, domain data events are required.

At the Application tier, the middleware is responsible to integrate and process incoming data from the Communication tier, delivering event streams with reliable location data to the Business Logic. This is performed based on a data fusion process that computes positioning data to provide accurate and reliable location data about the observed objects. Prior to the data fusion process, a set of location-based data for the same object has to be integrated. But after the data fusion process, a single computed position per object is provided, expressing a pre-processed set of events.

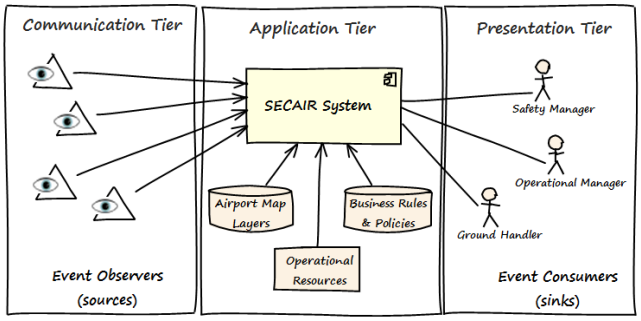


Fig. 3. A high-level view of the SECAIR system.

In order to be understood and processed, events need to be integrated with the business context, generating domain data events. Domain data events can be derived by mapping raw sensor data to domain concepts. For instance, domain data events correlate data of sensors located in one specific road segment, and evaluate speed limit infringements caused by a vehicle, or an alert caused by a tagged passenger for an unauthorized entrance into a restricted access area.

At the Application tier, there is logic that operates by interpreting a set of business rules to derive composite events from the events that have occurred. This means that one of the goals is to timely process generic data, not necessarily event notifications, tied together by spatio-temporal relationships, in order to perform a diagnosis based on existing business rules and organization policies. Such composite events are presented to decision-makers using the Client Application at the Presentation tier and might characterize a situation that is undesirable for the decision maker.

B. Communication Tier

The Communication tier ensures that the data observed by each device is timely transmitted to the Application tier at the server side. It is also aware about data communication requirements, including the wireless network required to cover the operational areas. The selected localization technologies acting as event observers are:

- The Stand-alone Global Navigation Satellite System (GNSS), used together with a Wi-Fi communication device, to collect and transmit, every second, the coordinates of the vehicle position;
- The Ultra Wide Band (UWB) system to provide immunity to multipath propagation and precision range measurement capability. The IEEE 802.15.4a UWB standard implements precision location measurements when the monitored objects are close to large metallic infrastructures;
- The Video Surveillance and Tracking System (VSTS) consisting of multiple video cameras installed at predefined locations to fully cover the target area. The video data collected by each camera is processed by the VSTS sub-system to detect, track and classify the foreground objects within the area of interest;
- The Radio Frequency (RF) localization system consisting of mobile devices and antenna units mounted in the area of interest. It measures the position of a mobile de-

vice attached to the observed object (e.g., passenger or staff) in the area of interest.

There is no single technology, which can provide satisfactory performance in all environments and scenarios; therefore, various localization technologies have to collaborate in order to deliver a flexible localization system, instead. Sensor data fusion will combine sensor data from different localization technologies to outperform any individual systems working alone.

C. Application Tier

At the server side, the Application tier is segmented into three conceptual areas. The middleware area to hold the Data Integration and Data Fusion, the Business Logic area holding the software components responsible for the system operational intelligence and the system operational database managed by Microsoft SQL Server 2008.

Middleware

The middleware is responsible to continuously provide a calculated position of each observed object to the Business Logic. The positions of the observed objects are continuously transmitted and are coherently integrated using data-fusion techniques to address multipath effects reduction and improve quality of location (QoL). This approach, besides reducing installation costs, also contributes to increase location accuracy, achieving a better coverage range with the same amount of equipment. [8]

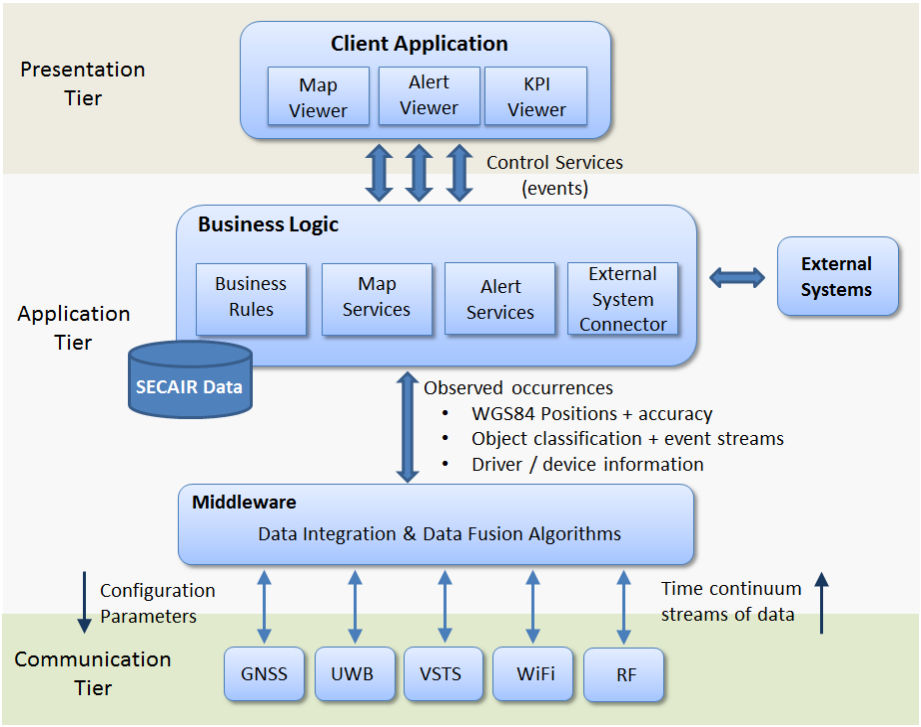


Fig. 4. The architecture of the SECAIR system.

Business Logic

For each data stream triggered by the Middleware, a set of actions are required to correlate additional data about the observed object with existing business rules and metadata about the spatial-context to determine domain events. This is accomplished at the Business Logic by four core software components. A short description of each software component is presented next.

- **Business Rules**, establishes the link between the definition and the execution of all business rules within the system, enabling organizational policies and the repeatable decisions associated with those policies, such as restricted area incursions, to be defined, deployed, monitored and maintained centrally at the server side. When changes occur at the business level, this service also assists in discovering the set of existing rules that are influenced by those changes. The Business Rules services interact with the operational database to store incoming events with the right classification.
- **Map Services**, this component is responsible to manage the spatial-context that characterizes the airport environment where the observed objects operate. Within the project scope, the airport layout is represented with a set of overlapped layers in a standard format [9]. To efficiently support the spatial database workload and the degree to which spatial functions are required, a geographic information system (GIS) engine was specifically designed. This GIS engine copes with challenging requirements related to scalability and real-time representation of multiple moving objects and dynamic changes to the spatial context, without

compromising the overall performance of the system. Depending on the nature of the detected event, the Alert Services will interact with the GIS to generate an alarm to be broadcast to each connected client application. A log record of all events is stored for historical data analysis purpose.

- **Alert Services**, for each business rule infringement a proper alert is generated by mapping location-based information related to the observed object (event) with domain concepts. This means that the data for each observed object has to be analysed to determine if a composite event (e.g., severe safety infringement) occurred or if a business metric needs to be updated. For each event being detected by the system, a semantic meaningful alert message will be triggered, with the corresponding relevance and severity risk. A dendrogram with weighted nodes is used to structure relationships between business indicators. Granular indicators are at the bottom (leaf) level and derived indicators (usually more aggregated) are at the nodes of the dendrogram. For instance, a business user can configure the system to inform about how many stand areas incursions were performed by a driver in a specific time period or day of the week.
- **External System Connector**, this software component is responsible for handling the interoperability with external systems, for instance, to collect data related with flight schedules, resources and assigned tasks. With such approach, location based data for each observed object can be coherently correlated with metadata from external sources, enabling the surveillance and track of events to be performed according to business logic/rules [10]. The Application tier, being responsible for implementing airport business logic, seeks ground for the coexistence and balance between the dual trends of the airport industry: increased demand for air travel and strengthened aviation safety and security [11].

Depending on the business rule being infringed, a specific event (alert message) is triggered to the end-users at the Presentation tier. This is done by creating a subscription offered as a public endpoint by the system. The Business Logic sends the requested data, either as a stream of updates (event-based queries) or as a chunk of current state data (instant-queries). The first are triggered on a certain event, e.g., an object moving into a specific area. The Business Logic can create an event subscription ("tell me about objects moving into a specific area") to be notified on that event ("an object moves into an area") and perform specified actions accordingly ("alert: object moved into restricted area"). This kind of subscription may be triggered often, or never, depending on how the event occurs. On the contrary, the result of an instant-query is always returned immediately and is not dependent on any event. This kind of query is useful to retrieve the current state of an object. For instance, "give me a list of all objects, which are currently in a certain area" or "tell me the current battery status of an object".

The data structure used to define each object position is presented by the class Position. The coordinates of the object position are presented as a point feature dataset that is used by the GIS engine to determine the location of each point feature (i.e., object location) over the airport layout. The airport layout is represented by a set of overlapped thematic map layers (also known as feature datasets, see Figure 5), some of them are polygon features representing operational areas with a predefined set of metadata to store specific business rules (see Figure 6). All layers have a common metadata structure; however, some polygon layers (e.g., Serviceroads, Stand, Taxiways, etc.) also have specific metadata attributes relevant for spatial context-semantic data analysis. These metadata are used by the Business Logic to enforce the application of predefined safety and business rules.

The GIS engine performs a topological point-in-polygon overlay operation to determine which points (i.e., IDObj) from the Position feature dataset are contained within the polygons of the airport layout. For each intersected layer, the Business Rule component interacts with GIS engine to check if any of the specified business rules is infringed. For instance, the Serviceroad layer includes attributes to specify the speed limit, category and status of each roadway segment within the airport, but the Stand layer just includes a status attribute to identify if a specific stand is open (i.e., with no parked aircraft), closed (i.e., with a parked aircraft) or deactivated (e.g., in maintenance). When a point is inside a specific roadway segment polygon, that information is passed to the Business Rules component to validate for operational rule infringements.

The QoL attribute is used to determine the accuracy of the reported position. This means that points reported with a QoL value higher than 10 m are labelled in red to indicate lack of accuracy in the IDObj position, QoL with values between 7.5 m and 10 m are labelled in yellow to indicate that the system is not able to assure the exact position of the IDObj and QoL with a value lower than 7.5 m are labelled in green to indicate that the IDObj position is accurate. The field tests performed so far were able to achieve position accuracy with a QoL between 0.5 m and 3 m.

A typical example of a safety rule infringement refers to speed limit, e.g., an object of type "vehicle" circulating at 30 Km/h within a roadway segment with a speed limit metadata of 25 Km/h will trigger a speed limit infringement alert message. In the same way any vehicle moving inside an open stand area will trigger a stand area incursion alert message. The IDObj provided within the Position data structure is used by the Business Rules to obtain more information about the IDObj (e.g., to which airport operator it belongs or if it is a priority object such as a "Follow-Me vehicle"). For each new position reported by each object, every second, the Business Rules might need to validate the IDObj business data before triggering an alert message.

As presented in Figure 4, additional business data about the IDObj are provided by the External Systems component. A data interoperability connection, established between the SECAIR system and the existing airport systems, enables the decision support capabilities of the SECAIR system to access some operational data (e.g., flight data and resource data) and correlate those data with the metadata from the airport layers and the data from the monitored IDObj. This computation is performed at the Business Logic in less than one second for each position reported by the middleware for each IDObj. In this way the SECAIR system is able to vali-

date for infringements to some business rules defined by each airport.

The system surveillance capabilities also includes collision avoidance to prevent, in real-time, ground damage to aircraft, equipment and potential injury to staff, operating within close proximity during ground handling operations.

To cope with such requirement (i.e., detection of the likelihood of collision trajectories), in the current version of the SECAIR system each object is represented as a point feature with a dynamic geofence around the object. For an aircraft, this geofence is designated as a clearance level, represented by a safety circle with a radius defined according to the specifications presented in ICAO Annex 14 [12] (i.e., wingspan and length of the fuselage). As such it is sufficient to generalize aircrafts to point locations instead of considering their real dimension. For vehicles, the geofence is designated as a protection area, corresponding to a rectangle at the front and another at the rear of the vehicle. The length of these two rectangles is determined by the vehicle category (e.g., Passenger Bus, High-Loader, Catering, Refuelling vehicle, etc.).

When any of these safety buffers (i.e., geofences) intersect, an alert for possible collisions between moving objects or other infrastructures is automatically triggered. For instance, when the vehicle protection area intersects the aircraft clearance a warning is triggered to indicate the driver to move to a safety distance. This functionality although in a preliminary version has been successfully tested when objects interact, namely for vehicles at the proximity of moving aircrafts or for interactions between Ground Service Equipment (GSE). For aircrafts parked at the stand area, the clearance level is not validated during ground handling operations. However, it will trigger a safety alert for any interaction of a parked aircraft with a moving aircraft (i.e., intersection of two clearance levels).

Collision avoidance between vehicles revealed to be a challenge difficult to accomplish because in most areas within the airport it is physically possible for a vehicle to move into any directions. Therefore, the specified protection area corresponds to a safe distance at the front and rear of each vehicle.

Operational Database (SECAIR Data)

For simplicity, the core data handled by the Business Rules software component are represented in Figure 4 by a single database. However, the physical implementation includes one relational database to store dynamic informational entities such as vehicles, operators, flights and aircrafts, and another to store the static airport cartographic layout, i.e., thematic map layers.

Both databases are managed by the Business Logic using the Microsoft SQL Server 2008, a database management system capable of dealing with business data and map features, describing the airport layout, within the same database.

In the SECAIR system, all thematic layers use the World Geodetic System 1984 (WGS84) as the spatial reference system in conformity to the specifications of the A-SMGCS manual [13] and comply with the ED-119 standard. The ED-119 standard defines the physical dataset requirements to develop the airport mapping. These include: geometry accuracy requirements, feature rules and descriptive (metadata) attributes. Since each layer is spatially referenced, they overlay one another and can be combined in a common map display.

The resulting geo-database consists of vector and attribute features. The vector features represent geometric feature instances that are classified as points, lines or polygons. As outlined in figure 4, each observed occurrence reported by the Middleware to the Business Logic is stored in the geo-database as new point features. This means that a new record with the object "Position" data structure is created within the object Position layer for each new position being reported.

The critical operational areas within the airport must have a polygon layer with specific metadata. Figure 5 presents an example of the type of metadata for the Stand layer (AM_PARSTANDAREA) and the set of metadata for a specific feature (i.e., stand named S14) within the Stand layer. Examples of critical polygon layers with metadata to validate safety infringements are: Runway area and Runway thresholds, Taxiways, Apron, Stand, Service road and holding lines just to mention some. In the SECAIR project there are fifteen critical polygon layers from a total of twenty nine layers used to characterize the airport layout.

D. Presentation tier

At the Presentation tier, the surveillance capability of the SECAIR system is presented to end-users in three different ways.

The Map viewer corresponds to a graphical layout managed by a GIS engine specifically designed to cope with two main requirements, namely to be operated by non-skilled airport stakeholders and to cope with airport stakeholder data processing needs for each spatio-temporal event. The Map Viewer represents the moving objects as colour coded point features with a timestamp and a set of descriptive data about the resources causing, for instance, a safety infringement; this might include data about the aircraft, vehicle, driver, flight data or layout of the area where the event occurred. The Map Viewer GIS engine is responsible to compute and represent in real-time (i.e., up to one second) the movements of all observed objects, computing simultaneously dynamic changes to the spatial context derived from daily airport business activities. Additional metadata (e.g., speed, logged driver, vehicle category) about each surveyed object are also provided.

The user can interact with the features of each layer by selecting, for instance, a specific stand and manually change its status, or obtain information about flights and tasks assigned to a specific stand area. It is also possible to verify which road segment is operational and check for traffic circulation rules that apply to the selected road segment (e.g., speed limit for different visibility conditions and directions of traffic flow) or analyse how many speed limit infringements occurred.

The Alert Viewer shows the corresponding textual description of alert messages in terms understandable by the end-user. This means that for each moving object causing an event, the Alert Viewer at each client application will present the alert messages contextualized with business semantic and

ordered by severity level. All alert messages have a start and an end time, plus a set of additional descriptive data related to each event.

The KPI Viewer presents, in a spatial dashboard, the values of key performance indicators describing how the business is performing. The correlations between KPIs are mapped in a dendrogram structure. Each node of the dendrogram carries some information needed for graphical visualisation of the data using size and colour coding.

IV. System Implementation

A. Communication protocol

We developed a protocol for communication between the hardware devices and the middleware (see Table 1). This protocol has the following characteristics:

- Independent of operating systems and hardware architectures
- Small packet size
- Clearly defined operations, even in complex use cases
- Full coverage of value range
- Easy to implement in various languages (complete implementation available in C# and C++).

Table 1. Communication protocol packet contents.

Bytes	Description
2	Magic "LP" = 0x4C50
1	ProtocolVersion = 0x02
8	Timestamp
16	Source ID
16	Message type qualifier (ID)
4	DataLength: Length of upcoming data field
variable	Data Area: Actual data field with payload

1) Timestamp

The timestamp is an unsigned 64 bit integer value storing the time when the position of the device is measured with nanosecond-precision.

2) Source ID

The source ID identifies the sensor or data system that originally created this message. Together with the timestamp, this can be used to generate an UUID compatible with RFC 4122.

3) Message type qualifier

The type qualifier is used to specify the type of content in a message. Only if the qualifier is known to a system, the message can be understood successfully. Messages with unknown qualifier should be ignored.

The message types include positional or environmental information, device health checking, firmware updates and several others.

B. Fusion of location data

Fusion of location data faces two main challenges. The first is associating objects tracked by the radio based sensors with those tracked by video technologies. The second is the fusion of those associated objects regardless of the technology it stems from.

To satisfy both requirements there are two interacting cycles. One is of high frequency and predicts positions. The other is of lower frequency and manages the association of objects.

The engine featuring these cycles merges data of widely varying quality into a single, continuous and seamless position track. Since the association algorithm knows which objects were tagged (e.g., staff), any observed objects without association can be considered non collaborative and originating potential issues. The discussion of the data fusion algorithm is outside the scope of this paper.

C. Control Service Implementation

Within the SECAIR project, control services are the building blocks to implement the business logic, enabling end-users to dynamically manage and interact with the target environment, changing the status of the business context as well as obtaining detailed information about moving objects and receiving automatic alert notifications about any safety infringements or incursions into restricted access areas. All location data are delivered by the platform to the presentation tier following an event-driven approach. End-user applications can subscribe to different events.

Some control services need to integrate with existing airport systems. Within the scope of the field tests at Faro airport, the external airport system used for demonstration is the Flight Information Data System (FIDS). The goal of such integration was to receive airport operational data in order to obtain, in real time, flight data.

We are assuming that location data corresponding to each object is provided by the underlying system. The quality of the data should be good and there should be no gaps (missing data). However, the services should be able to cope even with poor data.

Since we are working in a high security environment, delays or even crashes are to be avoided or at least handled properly. Logging relevant service-internal events and states is one of the basic ways to provide a reliable system. This can be extended by informing the system administrator on any encountered problems like failing services, starving or full buffers, network issues and configuration mistakes.

First of all, the very large system is broken down into various small components, each handling only a specific task. There is a basic differentiation between low level services that work with “raw data” and high level services that are based on those low level services.

Low level services

- Location: Current and historic positions of tracked objects. Event on new positions.
- Entity: List of all tracked objects along with some properties. Event on changed, added or removed objects.
- Map: List of all available areas along with some properties. Event on changed, added or removed areas.
- Storage: List of various data that are used and synchronized globally. Event on changed, added or removed data.
- Integration: A set of various separate services that are used to integrate the system with existing infrastructure. This includes, for instance, a service connecting to the airport flight management system and providing its data. Another included service will use these data to dynamically update the configuration of the alert services, in this case, allowing or disabling certain areas to be passed through by certain kinds of vehicles depending on whether or not an aircraft is moving through that area.

High level services

- Alerting: alerts are stored as a list of critical events along with some properties. Push notifications are triggered on added or changed alerts. Alerts are created by multiple separate services that use the above lower level services to create or change alerts based on certain business rules. For instance, by querying Location (position of an object), Entity (type of object), Map (areas) and Storage (stores permission table), an object of not-allowed type moving into a restricted area could create an alert.
- Multi-Tracking: based on the object position it is possible to activate in a new window the surveillance of the movements of the selected object. This is useful when there is a need to closely follow the movement pattern of one specific object.
- Collision Avoidance: the system can act preventively by determining if two objects (e.g., vehicle-vehicle or vehicle-aircraft) have a collision trajectory. This service is also relevant to assure clearance levels between aircrafts and to alert when a vehicle is not within a safe distance from moving aircrafts.
- Identify Resource: this service provides additional information about the selected object or map feature. In the first case, providing business data collected from the airport system, for instance, to describe the vehicle characteristics, obtain information about the logged driver or to correlate the current position of the vehicle with assigned tasks or flight schedules.
- Path Analysis: this service is used to draw a line as the object moves. The output enables the end-user to get a visual perception about movement patterns, understand patterns in specific time periods like rush hours or detect which vehicles are frequently out of predefined trajectories.

These control services, in general, correspond to actions performed by end-users individually; therefore, it is not feasible to introduce an overhead computation by implementing them at the server side. Some control services relate to business logic and need, therefore, to get access to business rules and business data to be able to actuate over predefined mobile devices or objects within the airport. Concerning the control service implementation let us describe the service usage, scaling and redundancy, connections via WCF and generic collection library.

1)Service usage

All services have to offer two basic things: a list of all data and notification of changes. Notification refers to the service actively informing the client of events. For instance, the location service will let all its clients know when a new position arrives. An alerting service could provide notice of new or changed infringements.

An event-based architecture is a pragmatic and reliable way to ensure fast response times, meaning that any occurrences are shown immediately (less than one second) and, as far as possible, dealt with automatically.

2)Scaling and redundancy

Unlike traditional monolithic systems, the services run as independent components and, as a matter of principle, are accessed via a network. In fact, it is no problem to run some of the services in a different part of the world if that would be advantageous.

One obvious advantage of this strategy is scaling. Since services run in parallel, just adding a few more machines will directly increase computation capability.

Also, losing a machine due to, for instance, network issues or hard disk failure, does not pose a problem. Another service will simply take over the work of the lost service for the time being.

To enable this architecture, data must not be stored locally on a machine. Instead, we are using distributed high performance databases connected via gigabit Ethernet.

3) Connections via WCF

The Application Programming Interface (API) is accessible via Windows Communication Foundation (WCF), one of the most common interchange formats. All connections are encrypted. Similar to the web service model much in fashion nowadays, instead of traditional large and complex interfaces, a simple and clean interface is used. However, in addition to traditional web services, we make use of events to decrease network overhead and provide real time updates.

Service interfaces basically feature only two kinds of data: data lists and events. Noticeably, they do not offer configuration to a client and are thus stateless. This is a feature common to web services and means clients have to specify a complete query at all times. The advantage is that the ser-

vices do not need to keep track of their clients, simplifying their code a lot and increasing performance generally.

The reasons for choosing WCF over competing technologies lie in its simplicity, reliability, interoperability and high performance coupled with many years of knowledge of using WCF "in the field". Remarkably, WCF is flexible enough to power both simple REST web services and complex state-aware multi-interface services. WCF can handle connections with clients written in different languages (e.g., Java) too, so interoperability with other system, for instance Linux, is a non-issue.

4) Generic connection library

Since all components use WCF, a generic library was established. It allows easily setting up servers and clients for any WCF service, no matter if a simplex or duplex connection is available. It also includes automatic endpoint creation for TCP/IP, HTTP (port 80) and metadata exchange.

Connection to the server is always fault aware. Any failure will transparently be encapsulated and provided to the client process, while avoiding any critical consequences. This means that all client services need to be aware of the fact that any connection may fail to work at any time, and cache their requests and data accordingly.

D. Monitoring

For easier deployment, we make use of WS Discovery. This allows services to publish themselves so that discovery-enabled clients can find them without any configuration. Since Discovery is a standard protocol, it is interoperable with other clients, services and proxies..

A supervision program is used to watch and control the state of all services. It ensures that all services are up and running, possibly restarting those that fail. It also displays the health state of the complete system to an administrator, additionally informing of important issues (e.g., a service failed permanently) by email or other means.

E. Graphical User Interface

The interaction between the software components at the Graphical User Interface (GUI) and the Business Logic are performed mainly through the lower level services to receive domain events. Based on that information, the end-user has the possibility to explore the information in more detail by executing the high level services.

V. USE CASE

In order to validate the SECAIR system, a system prototype for a pilot test is being installed at the Airport of Faro. For field tests, airport vehicles are provided and cameras from the VSTS system are installed. The existing Wi-Fi network is strengthened at specific operational areas in the airport.

Some vehicles will be equipped with an onboard unit, a touch screen display and a radiofrequency reader to automate the driver authentication procedure. All personnel participating in the field tests will receive a radiofrequency card. The association of the driver with the vehicle is performed automatically each time the card is read by the radiofrequency reader installed in the vehicle. At least two client applications are deployed, one situation room to the Apron Control Centre for airport operators and a second situation room to support other airport stakeholders (e.g., Ground Handler).

Figure 5 outlines the airport air side areas selected for the site tests. Three operational areas were considered: the passenger terminal (Boarding Gates 01 and 02) for indoor test scenarios, adjacent indoor-outdoor transition area (to demonstrate ability to track targets moving from indoor to outdoor and back) and the Apron area adjacent to stands 14 and 16 for outdoor test scenarios.

Indoor scenarios (see Table 2) include: zone intrusion detection, target tracking and left behind luggage. Indoor-outdoor transition areas include the surveillance of people at the boarding gates for (dis)embarking procedures. Indoor scenarios reflect operational procedures related mainly with the observation of people and baggage in restricted access areas within the passenger terminal:

- Traceability of a person at the boarding gate area;
- Localization capability of the SECAIR system in the transition area from passenger terminal into restricted access areas (outdoor);
- Location obtained by fusion of data obtained from the following technologies: VSTS and RF.

Outdoor scenarios (see Table 2) cover an area defined by 130x130m of the apron comprising Stands 14 and 16. The outdoor scenarios reflect operational procedures related mostly with the observation of vehicle movements and operational procedures related to parked aircraft assisting tasks:

- Traceability of vehicle and driver at the Apron area;
- Automatic detection of drivers without driving permission / not logged;
- Location obtained by fusion of data obtained from the following technologies: VSTS, GNSS and UWB.

As presented in Figure 5, after a successful login to the SECAIR system, each connected client visualizes the current status of the airport layout as a collection of overlapped themes, each one representing a specific operational area within the airport environment. These themes are managed by the Map viewer, forming the background context over which the observed objects are represented as point features. All thematic layers were provided by the airport authority in a standard format as shape files.

The selection of the "Maps" option at the sidebar, enables the end-user to dynamically access each feature within a specific layer. For instance, to select a feature (e.g., Stand 14) from the AMD_PARKSTANDAREA layer, which is one of the layers used to accurately represent the apron layout, the user just has to select the corresponding polygon at the Map viewer (or navigate to the corresponding layer at the sidebar and select the feature he is interested in). As presented in Figure 5, the apron folder is a logical folder used to semantically group layers; therefore, it is possible to group layers with different geometries (e.g., points, lines and polygons) in the same folder.

Some layers are used just to help end-users to get contextualized with the airport area they are visualizing. These layers are not used by the Business Logic to compute domain events. At this phase and during the field tests only five layers were used to compute domain events:

- AMD_TERMINAL, layer representing the passenger terminal indoor areas,
- AMD_PARKSTANDAREA, layer representing the geometry of all stand areas. Figure 6 presents a print screen of the metadata defined for the layer and the metadata defined for stand 14;
- AMD_ APRONELEMENT, layer representing the geometry of the Apron area;
- AMD_ TWYELEMENT, layer representing the geometry of all taxiways;
- AMD_ SERVICEROAD, layer representing the geometry of all roadways.

When a feature is selected, besides visualizing the feature geometry at the Map viewer, the end-user can also access the metadata defined for the selected feature. Metadata common to all features are specified at the layer level, for instance, to indicate which categories of vehicles are allowed to circulate. These metadata might be duplicated if there is a need to include values for normal and low visibility operations. Metadata specific to a feature are specified individually to each feature within a layer, for instance, to cope with temporary changes to one feature or to cover business rules, which apply only to a specific feature or set of features.

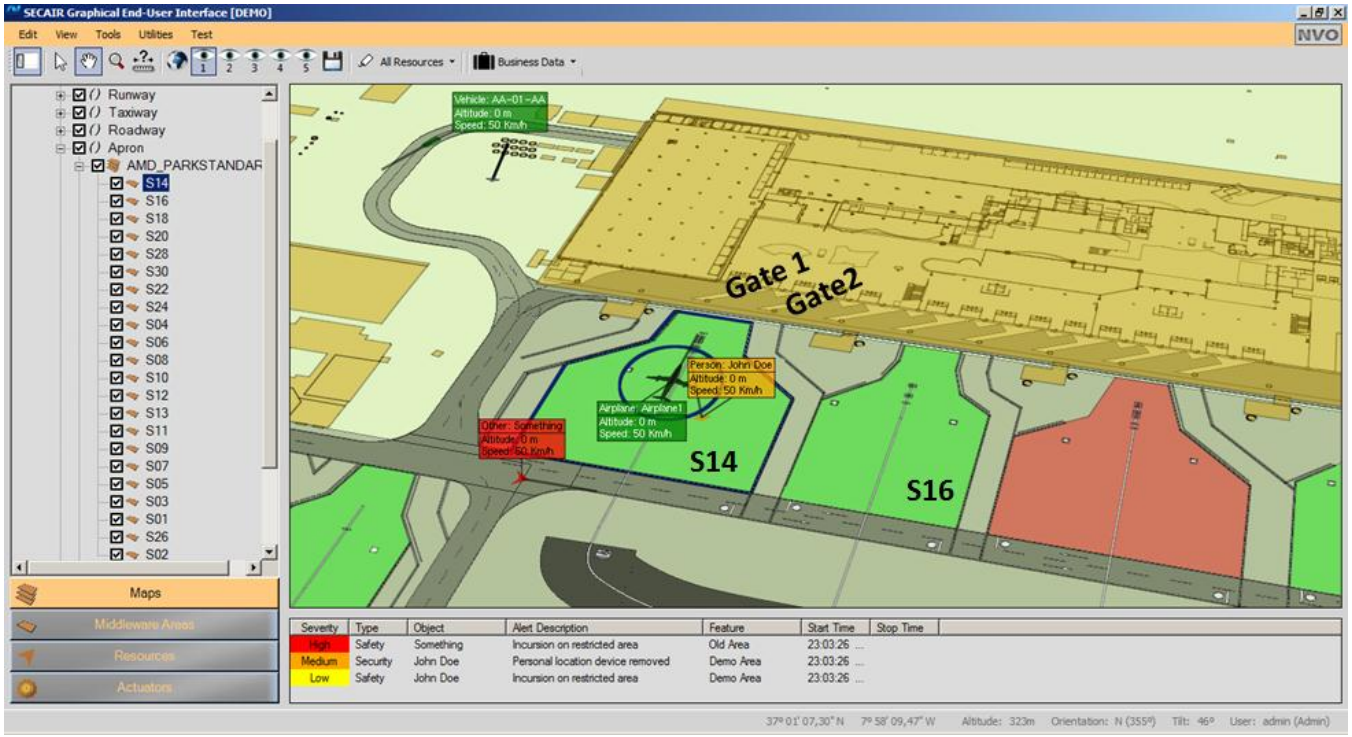


Fig. 5. Airport layout of the areas selected for the specified scenarios.

In SECAIR, most of the business rules related to airport operational areas have a strong spatial dependency with the metadata defined for polygon features, meaning that in a certain instant the position of the observed object (represented as a point feature) can trigger multiple events. When the GIS engine performs a topological point-in-polygon overlay operation, the current point position of the object is analysed against all intersected layers.

For each intersected layer, the Business Rule component will check if any of the specified business rules is infringed. Since the system operates with a collection of overlapped themes, classified accordingly to their relevance, when monitoring for business rule infringements the algorithm orders the resulting events according to their severity level. This means that, at the Presentation tier, domain events occurring in critical areas are visualized with a higher priority in relation to those occurring in less critical areas (e.g., a runway incursion is more severe than a stand area incursion). When multiple events are detected, the Map viewer only represents the most severe event. The other events are listed at the Alert viewer, enabling the end-user to have a good perception about the sequence of events caused by a specific person/vehicle.

The validation of spatio-temporal events gets even more complicated because for each object movement the system has to perform a set of topological point-in-polygon compu-

tations to determine whether a given point in the plane lies inside, outside, or on the boundary of each intersected polygon features. If we consider that at each instance it is possible to have multiple objects and that a new position is triggered, even when a vehicle is not moving, it starts to be very demanding to comply with all the requirements.

For each event, the Business Logic uses the metadata provided by each layer for location-awareness purposes and formulation of the alert message to be visualized by the end-user at the Presentation tier.

The design approach was to provide, in a single screen, the user with all the data relevant for him to take informed decisions. Therefore, the Map viewer corresponds to a spatial dashboard where all observed objects are represented as colour coded point features and the airport layout as the background spatial context, which is used to semantically transmit implicit information about the status of some operational areas.

For instance, an occupied stand is represented with a green polygon, but if it is under maintenance, the same polygon is visualized in grey. In both cases, vehicles are allowed to circulate inside the stand area. In the first situation, vehicles are allowed to enter the stand area to assist a parked aircraft and, in the latter, because the stand area is deactivated. But when a stand is operational with a parked aircraft, as soon as the information about blocks-off is reported to the SECAIR system, the operational status of the stand area automatically changes to cleared, causing the colour of the polygon to change to red to signal a restricted access area where vehicles are not allowed to circulate or park.

Such visual clues transmit valuable insights to the end-user at the control centre. To simplify the identification of events, moving objects can be visualized with a colour coded label to easily transmit which objects are causing infringements from those operating as expected, i.e., labelled with a green colour. Once again, a colour code is used to differentiate less severe (labelled in yellow) from severe events (labelled in red).

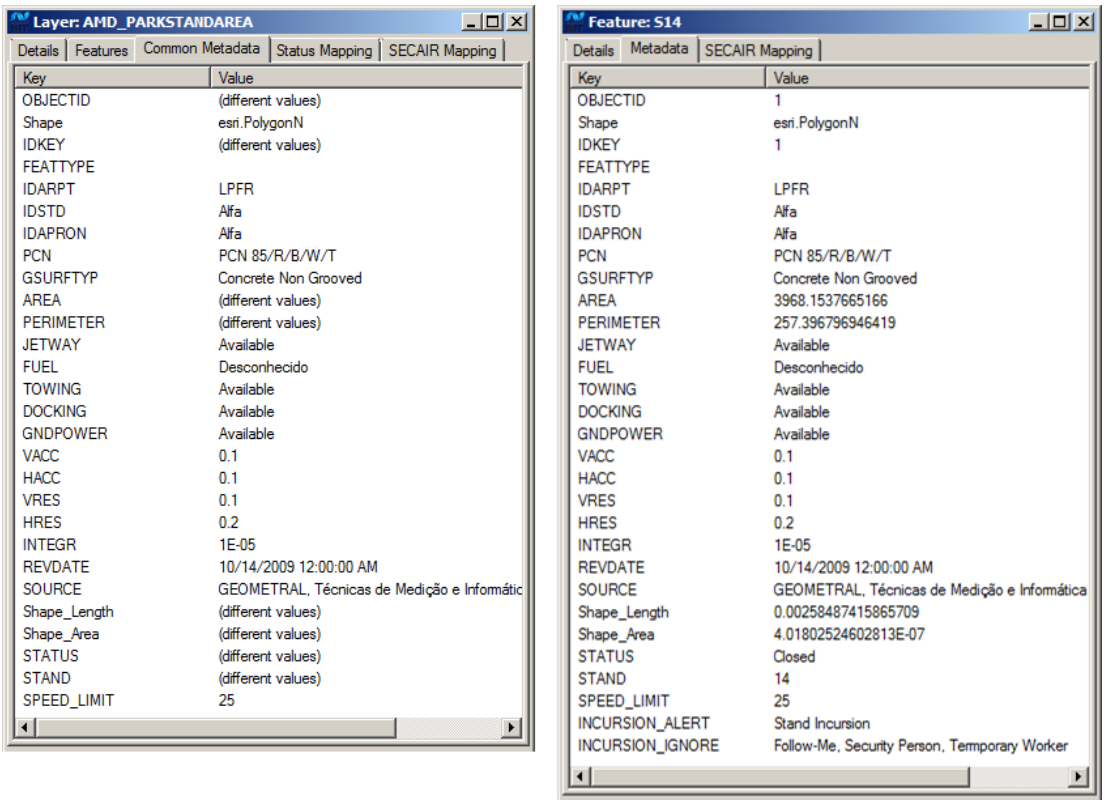


Fig. 6. Metadata assigned to the Layer level and metadata assigned to a specific feature within the layer.

All events are logged and represented at the Alert viewer with information about the event (e.g., severity, type, operational area, instant, etc.). This feature is useful to support decision makers in case there is a need to check for a specific event reported during the end-user shift. The reported information is automatically filtered based on the end-user profile, meaning that the list of events reported in the Alert viewer relate to resources from the airport stakeholder. Only the airport authority (e.g., Safety Manager) has access to all reported events.

The KPI viewer presents a set of predefined indicators specified for each of the operational scenarios defined in Table 2.

Table 2. Operational scenarios specified for the field test.

Num.	Scenario Name
Outdoor Safety (OSA)	
OSA.01	Surveillance of vehicle movements within a stand area
OSA.02	Collision avoidance support service
OSA.03	Aircraft ground movement tracking
OSA.04	Obstruction of an operational stand area
Outdoor Security (OSE)	
OSE.01	Detection of zone intrusion by unauthorized vehicle
OSE.02	Personnel tracking at the apron area
Indoor Safety (ISA)	
ISA.01	Working zone intrusion by unauthorized person
Indoor Security (ISE)	
ISE.01	Left luggage detection
ISE.02	Indoor-outdoor personnel tracking

The description of each operational scenario follows a template emphasizing relevant issues from the airport stakeholders' point of view. Besides a unique identifier with a semantic meaning for each scenario, the template also covers the following items:

- Name of the scenario, pointing out concerns from the perspective of airport stakeholders;
- Classification of the scenario, addressing environmental influences (indoor/outdoor) and type of events (Safety/Security);
- Technical constraints and a list of key indicators captured by the scenario to measure its impact or relevance;
- List of actions to be performed by each intervenient actor to test the specified scenario;
- Identification of the expected results for each scenario, defining the behaviour of the SECAIR system.

The net outcome of the SECAIR system will be the creation of improved or new services to increase security and safety of airport (or other critical infrastructures). In order to deliver such services, the SECAIR system needs to interoperate with the existing airport system to acquire business information relevant to properly manage domain events, namely data related to:

- Flight schedules, to align the system actions with the flight information scheduled for each stand, tasks allocated to assist the aircraft and reflect any last minute change to what was planned;
- Tasks assigned to staff, including ground handlers ;
- Staff, data relevant to support the system operational intelligence, including data about the airport operator he works to, role performed within the airport and specific data related to their airport driving licence;
- List of all resources (e.g., vehicles and equipment) operating at the airside area;
- Access to information about the main characteristics of the aircrafts operating at the airport.

VI. RELATEDWORK

The surveillance of surface movement events using ground surveillance is challenging for several reasons. First, because of the huge amount of demanding requirements that have to be validated to protect passengers, staff and critical infrastructure from serious security or safety infringements. Second, the quantity and quality of available surveillance is often poor. In an airport environment, unless the surveyed objects are equipped with a transponder, surveillance is based on surface movement radar (SMR) returns only. However, these solutions are extremely expensive to purchase and operate, and are subject to masking and distortion in the vicinity of airport buildings, terrain or plants [14].

The extensive deployment of satellite system and air-to-ground data links results in the emergence of complementary means and techniques. Among these, ADS-B (Automatic Dependent Surveillance-Broadcast) and MLAT (Multilateration) techniques are the most representative [15]. However, current radar based systems have many problems to track surface targets, especially in very dense traffic areas, such as the apron area. On the other hand, most aircrafts turn-off their transponders after landing, compromising their identification possibilities. This means that such systems will not detect non-cooperative vehicles or aircrafts that are not equipped with such a transponder. Therefore, there is a strong demand for a new sensing technology, in particular for smaller/medium airports. Such sensors include near-range radar networks, Mode 3/A, S or VHF multilateration, CCTV systems with video analytics [16], magnetic flux sensors or D-GPS installed in vehicles [17]. Although none of these sensors is individually able to meet all user requirements for airport surveillance, the fusion of the information they give can lead to an acceptable solution.

The Airport Surface Detection Equipment, Model X (ASDE-X) system [18], adopted in the United States, provides precise time-stamped position as required for its primary mission of improving situation awareness. ASDE-X surveillance is based on plot-level fusion of multiple complementary sensors, providing air traffic controllers with highly accurate, real-time position and identification information of all aircraft and vehicles on the airport surface. The system accepts and fuses primary SMR returns from objects on the airport surface. Any available report from ADS-B or secondary surveillance radar is also considered in estimating and reporting the fused track position. But the cost of this solution is high.

But, for the airport to function, it needs to have the right mix of these independent systems, and these systems need to cooperate with each other. At such level of interdependencies the airport system pursues a common goal for a set of stakeholders according to their concerns, which cannot be achieved by these entities individually. This means that an airport is a system-of-systems bounded by and understood through the identification and analysis of existing systems, the knowledge of operational procedures and regulations and organizational issues influencing the system of interest. In recent work [19], it is already possible to find algorithms addressing the very stringent integrity requirements to support complex event processing (CEP).

The CEP paradigm arose as a solution for critical environments, where there is a need to fuse a huge amount of information in order to detect events of interest in the company workflow [20]. Several proposals to fusion this company information have followed a CEP approach in many fields

like financial transactions [21], business decisions [22] or RFID-based services [23]. The CEP paradigm has been used in the present work as the mechanism to orchestrate the contextual information inferred by other parts of the system. This means a new approach to aggregate and fusion of situation-awareness information about the observed objects within the organizational context in which they occur. Event-driven organizations are expected to react quickly to changes in their environment. Thus, their decision support systems should be driven by the same transactional events that keep the business operating. Such approach focuses on developing on-board applications that allow a vehicle to infer its role in a scene by taking as input, data from different sensors of the vehicle and other external data sources [24]. Such sensing can provide comprehensive information about the vehicle context, extending the surveillance capabilities of the system not only to perceive the situation context of a vehicle, but also the whole context related to its role within the environment where it operates.

In the literature, it is possible to find research projects (e.g., Airnet at www.airnet-project.com, 2004; ISMAEL at www.ismael-project.net, 2006; EMMA at www.dlr.de/emma, 2008; AAS at www.aas-project.eu, 2010, and LOCON at www.locon-eu.com, 2011), with different technologies that have been developed and successfully tested for ground movement detection, providing actionable data with a high degree of certainty or as a cost-effective solution.

Besides the identified technological related issues, the interactions between system components and between humans and software applications are also subject of interest within the academic community, in particular research on discrete event monitoring [25], [26]. These references consider safety requirements as control structures that restrict system behaviour at meta-model level. They propose a framework for interface control systems. In this framework, functional requirements and safety requirements are separately formalized as interface automata and controlling automata respectively.

According to their approach, requirements include two subtypes: functional requirements and safety requirements, which are requirements about the safe operation of the target system. There are two common causes of the changes to safety requirements. First, safety requirements may change at design-time. Second, safety requirements may change post-implementation. Some safety requirements are unknown before the system is developed and used in real environment. The requirement specification concerns that guided the design of SECAIR system in compliance with end-user functional requirements is in line with the methodology adopted in [27], to fill in the gap between the evolution of safety requirements and traditional verification process.

A similar approach is also conducted in [28], the motivation for this work emerges from multi-agent systems as a paradigm for developing complex, software intensive systems in real world scenarios. According to the multi-agent paradigm, agents are able to interact and to reply to events triggered by their environment. In this work, a detailed presentation of the Ptolemy II tool is presented to support the design and verification of resource constrained in embedded systems. The proposed solution allows the modelling of functional and dependability requirements separately. The functional model is described in terms of labelled interface automata, an action-oriented approach that considers not only the control flow, but also the information flow (input/output actions). Safety and security constraints are specified using controlling automata. It also applies model checking techniques in order to automatically generate a compliant model that will satisfy the dependability requirements.

Within the SECAIR project, one of the technical requirements is that all location–based technologies are coherently integrated using advanced data-fusion techniques in order to reduce installation costs and to address multipath effects reduction. The project fusion techniques operating with high-performance GNSS systems and improved radio based tracking, combined with video based technology will enable the accomplishment of an automatic and reliable prediction of events related to safety infringements. This broad level of integration extends the state-of-the-art for the surveillance of airport surface traffic, enabling unique automated decision support capabilities, with new context aware services that have not, thus far, been available.

Very concretely, the results of the SECAIR project might contribute to some other initiatives and product developments [29] [30] [31] [32] [33].

VII. CONCLUSION

The SECAIR system improves context awareness and controllability via the integration of localization technologies. By integrating multiple sensor technologies, the system delivers a comprehensive picture of ground operations, increasing the controller situational awareness and improving the airport safety.

In the SECAIR, multiple occurrences can be combined from different localization technologies to provide insight into business operations, enabling in-time decision making. Within the SECAIR project, one of the technical requirements is that all localization technologies are coherently integrated using data-fusion techniques in order to reduce installation costs and to address multipath effects reduction. The main objective is to develop new context aware services based on an innovative solution integrating high-performance RF tracking combined with video recognition technologies and mobility management in a middleware platform.

SECAIR is designed as heterogeneous sensor data fusion system architecture, covering the surveillance of non-cooperative resources and functionalities for continuous control of all ground movements within the apron area. A special attention is given to the environment of the system, in particular to information flowing from and to the system. The properties of the system components, as well as the relationships between them, are core elements for the analysis and design of the SECAIR architecture. The project takes lessons learned from previous projects in relation to techniques for multi-target, multi-sensor tracking, responding to very important security and safety issues in airport environments. The software components of the SECAIR system are

being tested and a field trial is planned to take place at Faro airport during the last quarter of 2013, with a full evaluation of the results to be done during the first half of 2014.

ACKNOWLEDGEMENTS

SECAIR (ref. E6030) is an R&D project, partially funded under the EUROSTARS program. The authors acknowledge the collaboration of the remaining partners of the project. It is also acknowledged the funding of FCT through the PIDDAC program and of Agência de Inovação.

REFERENCES

[1] G. Pestana, A. Casaca, P. Reis, S. Heuchler, and J. Metter, "Management of Mobile Objects in an Airport Environment", Proceedings of the Second International Conference on Mobile Services, Resources, and Users – Mobility 2012, ISBN:978-1-61208-229-5, pp. 55-59, Venice, Italy, October 2012.

[2] Eurocontrol, "Operational Concept and Requirements for A-SMGCS Implementation Level 2. Ed. 2.1", 2010.

[3] G. Pestana, N. Duarte, I. Rebelo, and S. Couronné, "Adressing stakeholders coordination for airport efficiency and decision-support requirements". Journal of Aerospace Operations (JAO11), 2011.

[4] SECAIR Deliverable 1.1, "Definition of requirements and operational scenarios". Technical deliverable from SECAIR project, Eurostars Program, Brussels, December 2011.

[5] Luckham, David C., "Event Processing for Business: Organizing the Real-Time Enterprise". Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

[6] M. Ayres Jr. et al, "Safety Management Systems for Airports: Guidebook. Volume 2, ACRP REPORT 1, ISBN 978-0-309-11798-2, 2009.

[7] ISO/IEC/IEEE 42010:2011(E), "Systems and software engineering — Architecture description". 1st ed., Institute of Electrical and Electronics Engineers, Inc., 2011.

[8] W. Schuster, J. Bai, S. Feng, and W. Ochieng, "Integrity monitoring algorithms for airport surface movement", SpringerLink, vol. 16, N. 1, pp. 65-75, 2012.

[9] EUROCAE, "ED-119B: Interchange Standards For Terrain, Obstacle, And Aerodrome Mapping Data". EUROCAE ED-119B / RTCA DO-291B, 2011.

[10] M. Agaram and C. Liu, "An Engine-independent Framework for Business Rules Development". In Proc. of 15th IEEE International Enterprise Distributed Object Computing Conference. 2011.

[11] N. Subbotin, "Development of an Airport Ground Vehicle Runway Incursion Warning System". DOT/FAA/AR-11/26, 2011.

[12] IATA, "Airport Handling Manual (doc. AHM 630)", 30th ed., 20130.

[13] Eurocontrol, "A-SMGCS, Advanced - Surface Movement Guidance and Control System Manual", International Civil Aviation Organization, Approved by the Secretary General and published under his authority, Montreal, 2004.

[14] T. T. W. Hu, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors", IEEE Transacions on Systems Man, and Cybernetics-part C: Applications and Reviews, pages 334 - 351, 2004.

[15] A. Soto, P. Merino, and J. Valle, "ADS-B integration in the SESAR surface surveillance architecture", Proc. of Enhanced Surveillance of Aircraft and Vehicles (ESAV), pp. 13 -18, 2011.

[16] P. A. Vela, et al., "Visual Monitirong of Airport Ground Operactions", proc.of IEEE/AIAA 28th Digital Avionics Systems Conference, 2009.

[17] H. Gao, et al., "Safe airport operation based on innovative magnetic detector system", Institution of Engineering and Technology, Transp. Syst., Vol. 3, Iss. 2, pp. 236-244, 2009.

[18] E. D. Yuan and M. Watton, "An event-driven service oriented architecture for Space Command and Control decision making," 2012 IEEE Aerospace Conference, pp. 1–9, Mar. 2012.

[19] F. Terroso-Sáenz, M. Valdés-Vela, F. Campuzano, J. A. Botia, and A. F. Skarmeta-Gómez, "A complex event processing approach to perceive the vehicular context," Information Fusion, Sep. 2012.

[20] J. Dunkel, A. Fernández, R. Ortiz, and S. Ossowski, "Event-driven architecture for decision support in traffic management systems," Expert Systems with Applications, vol. 38, no. 6, pp. 6530–6539, Jun. 2011.

[21] K. Mani Chandy, "Event-Driven Applications: Costs, Benefits and Design Approaches". California Institute of Technology, 2006

[22] V. Pillac, C. Guéret, and A. L. Medaglia, "An event-driven optimization framework for dynamic vehicle routing". Decision Support Systems, vol. 54, no. 1, pp. 414–423, Dec. 2012.

[23] G. Cugola and A. Margara, "Complex event processing with T-REX," Journal of Systems and Software, vol. 85, no. 8, pp. 1709–1728, Aug. 2012.

[24] T. Waldron, S. Corporation, and E. Syracuse, "Detecting Airtport Surface Movements using Ground Surveillance", Digital Avionics Systems Conference, 2009. DASC '09. IEEE/AIAA 28th 2009.

[25] Zhe Chen and Gilles Motet, "Towards Better Support for the Evolution of Safety Requirements via the Model Monitoring Approach". In Proceedings of the ACM/IEEE 32nd International Conference on Software Engineering (ICSE 2010), pp. 219-222

[26] Zhe Chen and Gilles Motet, "System Safety Requirements as Control Structures". In Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC), pp. 324-331, 2009.

[27] Gianina Homoceanu and Michaela Huhn, "Tool support for agent-based systems in Ptolemy". In Proceedings of the International Workshop on Security and Dependability for Resource Constrained Embedded Systems (S&D4RCES), 2011.

[28] Y. Liu, I. Gorton, and V. K. Lee, "The architecture of an event correlation service for adaptive middleware-based applications", Journal of Systems and Software no. 81, pp. 2134–2145, 2008.

[29] PPS-Locate and PPS-Airports, Brand of PPS GmbH, Nordostpark 93, 90144 Nuremberg, Germany, ® 2013.

[30] EUROCONTROL SWIM, part of the European ATM Master Plan, Contract between BIJO-DATA and European Organisation for the Safety of Air Navigation, Brusselles-Sesslach, June 2012.

[31] EUROCONTROL SESAR, Single European Sky ATM Research, Council Regulation (EC) No 1361/2008, on the establishment of a Joint Undertaking ("SJU") to develop the new generation European air traffic management system (SESAR), 2008.

[32] V. Mehta et al., "Decision Support Tools for the Tower Flight Data Mananger System", Integrated Communications, Navigation and Surveillance Conference (ICNS), pp. I4-1 to I4-12, 2011.

[33] J. Metter, "EFSUES - Energy efficiency increasing system for controlling and monitoring of airport apron", TNA XI-1/2012, SMI GmbH, Untere Burgbergstrasse 382, D-98663 Heldburg, Germany, Heldburg, 2012.

A Holistic Approach to Energy Efficiency Systems through Consumption Management and Big Data Analytics

Ignacio González Alonso
University of Oviedo
Oviedo, Spain
gonzalezaloignacio@uniovi.es

María Rodríguez Fernández
University of Oviedo
Oviedo, Spain
rodriguezfmaria@uniovi.es

Juan Jacobo Peralta
Andalusian Institute of Technology
Málaga, Spain
jjperalta@iat.es

Adolfo Cortés García
Ingenia S.A.
Málaga, Spain
adolfo@ingenia.es

***Abstract*— Improving the energy efficiency is one of the most effective ways to increase supply security and reduce Green House Gasses emissions. Furthermore, the increased cost of energy has encouraged the development of new technologies that allow the efficient use of them, such as monitoring the final users energy demand; hence, it is possible to have a more efficient consumption behavior without lowering the threshold of comfort that consumers are used to. The Smart Home Energy project makes a profitable use of these technologies by allowing the final user to manage, control, plan and, in most cases, reduce the electric bill. To facilitate the bidirectional interaction between the customer and the devices integrated in the smart home communication network it is necessary to follow a holistic approach. This proposal aims to go a step further by using the massive consumption datasets to predict future energy behaviors, offering personalized recommendations and developing a customized "consumer energy knowledge" for every home, through heavily processed Machine Learning algorithms.**

Keywords*— *Digital Home; Energy Efficiency; Smart Metering; Cloud; Big Data Analytics

I. INTRODUCTION

From a technical point of view, a home is an extremely complex system with many uncertain variables, which can be influenced by environmental conditions and human behavior.

In recent years, the appliances that are being incorporated to the households are reaching higher and higher levels of energy efficiency. However, these contributions are still not enough in the current energy scenario, where external energy dependence and indefinite rising prices question the profitability of these devices compared to their useful life-spam and their necessary initial investment.

In addition, these facts cause that energy saving measures and common recommendations (energy-efficient light bulbs, electrical appliances A++, awareness campaigns, etc.) are increasingly ineffective in achieving a significant reduction of the energy costs, stating the need of more advanced strategies.

Therefore, the inclusion of management and control systems can represent an adequate line of action to increase the energy efficiency at household level. It permits the quantification of the home energy demand, recording the consumption values and characterizing personalized profiles of energy patterns. Based on this information, a set of recommendations can be generated, which would modify the consumer behavior and ultimately result in reducing the electric bill. Other complementary visual tools can help also to analyze the consumption and consequently to reduce it.

Due to the amount of data that these systems has to handle is large and expected to keep growing in the future, the architecture work presented at ICSEA 2012 [1] has been revised and extended with the description of an operational system, where the addition of new Big Data based components is the key, in order to give the necessary support to a massive set of Digital Homes.

The paper is organized as follows: in Section II, the State of the Art is described. The proposed solution is explained in Section III. A brief extension of the work focusing on the exploitation of the data obtained by the system is done in Section IV. In the last section, conclusions are drawn and future work is indicated.

II. STATE OF THE ART

The integration of various automation and control technologies in the domestic environment is called "Digital Home" [2]. This term is not only applied on the domestic tasks performed by smart appliances, but also aims to cover specific customer needs of personal assistance, education and/or entertainment as well as security and surveillance.

One of the main objectives of a Digital Home is to design an efficient Energy Management System (EMS). Therefore, an appropriate starting point is to study the features and functionality of Building Management Systems (BMS) [3] such as tasks related to the Heating, Ventilation, and Air

Conditioning control, a correct monitoring of lighting, allowing a control of consumption and its associated costs.

In terms of architecture, BMS has a similar structure and characteristics to those described on the mentioned home environment: hardware components as sensors, computer processing power, user interfaces (smartphones, PCs, tablets, etc.) as well as means of transmission. However, neither control of electrical appliances and service robots, nor the integration of smart metering devices, are covered by BMS.

In the last decades, the integration of Distributed Energy Generation Units (DEG) at the household level constitutes the latest addition to the in-home EMS [4]. There are various types of renewable technologies available to install at homes, but solar and wind power energy sources are the most popular ones. Due to unpredictable weather fluctuations, it is challenging to incorporate this type of energy sources to the home systems and elaborate accurate energy productions, hence reliable prediction models are a main milestone to make these installations profitable. Weather data needed to make the consumption predictions are highly variable in a worldwide approach, because of its different sources (meteorological services and institutions), formats, and protocols. Short time meteorological predictions also change frequently, where an hourly interval is usual to get new predictions for the next 72 hours, meaning that done predictions also could be changing hourly.

After integrating these DEG Units, the more integrated and synchronized the home energy systems are, the more energy savings are achieved due to self-production consumed energy. Additionally, these systems are also acceptable to help energy operators to get easily equilibrium between the energy demand and supply curve. Finally, due to the decentralization of the energy system, the losses inherent to its medium and long distance transportation will be significantly reduced.

From a physical standpoint, the different elements that compose the EMS, e.g., appliances, domotic control elements or renewable energies, may interact with each other, exchange heat and an influence in the in-doors temperature and humidity, changing comfort conditions. For that reason, the accurate quantification and statistical analysis of all variables involved in the heat exchange together with its possible interactions have to be made in order to propose real energy-efficient alternatives within the comfort conditions predefined by the user.

The domain of smart homes develops fast but still many restrictions have to be faced: the high cost of certain systems, storage and processing limitations, etc. Nevertheless, the most important limitation to solve is the lack of true interoperability between systems of different manufacturers, which makes them to work independently. That could result at the end in a duplication of the performed tasks. So, it is essential to communicate those devices with each other in a complementary way, i.e., sharing the services to study what is happening around them and take decisions accordingly.

The use of an open, extensible and modular protocol would resolve that requirement. An open platform to be considered is UniversAAL [5], supported by the European Commission, which has taken promising steps by aiming to design, develop, evaluate, standardize, and maintain a common service platform for Ambient Assisted Living.

Covering a wider field, the standard DHCompliant (DHC) [6] communication protocol at the application level of TCP/IP stack offers a cost-effective solution regarding other protocols, such as technological and brand independence, savings in solution investment and control and simultaneous management of service robots and smart appliances.

Regarding the storage and processing of data, and in contrast to other researches [7][8], all business logic will be in the Cloud. Although it is usual to allow remote control, the exploitation of the data is performed locally [9]. Commercial solutions, which are typically linked to a specific product of smart metering, which allow local monitoring of a home, do not allow relating information of several houses to extract information collaboratively.

An energy consumption metering infrastructure generates large amount of data per home and the quantity increases proportionally with the number of houses sending data simultaneously. It should also be taken into account the large number of users requesting information. In an acceptable time response, the system must be able to acquire, store and process large amount of data, and scale horizontally. In the gap between the very expensive solutions of specific hardware supercomputers and enterprise servers, Big Data techniques are the key for achieving that goal [10].

Moreover, it provides more capabilities to work with this large amount of data. It should be mention the integration with Machine Learning algorithms, generation of statistics, capacity of mixing varied data such as internal data (measurements) and external data (climatology, open data, etc.). Although there are precedents of using machine learning techniques to improve energy efficiency [11], to the best of our knowledge, they are not been applied in a collaborative way, i.e., using the data collected from all houses that are part of a larger network, known as Smart Grid.

III. PROPOSED SOLUTION

The presented solution is framed within the Smart Home Energy (SHE) project [12]. The architecture of the system was designed following a necessary holistic approach that takes into account all the elements involved in a smart domestic environment, such as electrical appliances, lighting, people or external conditions, in order to model and analyze all contributions and interactions.

As it is graphically shown in Figure 1, the main components of the architecture, which will be explained in the following subsections, can be grouped according their physical situation:

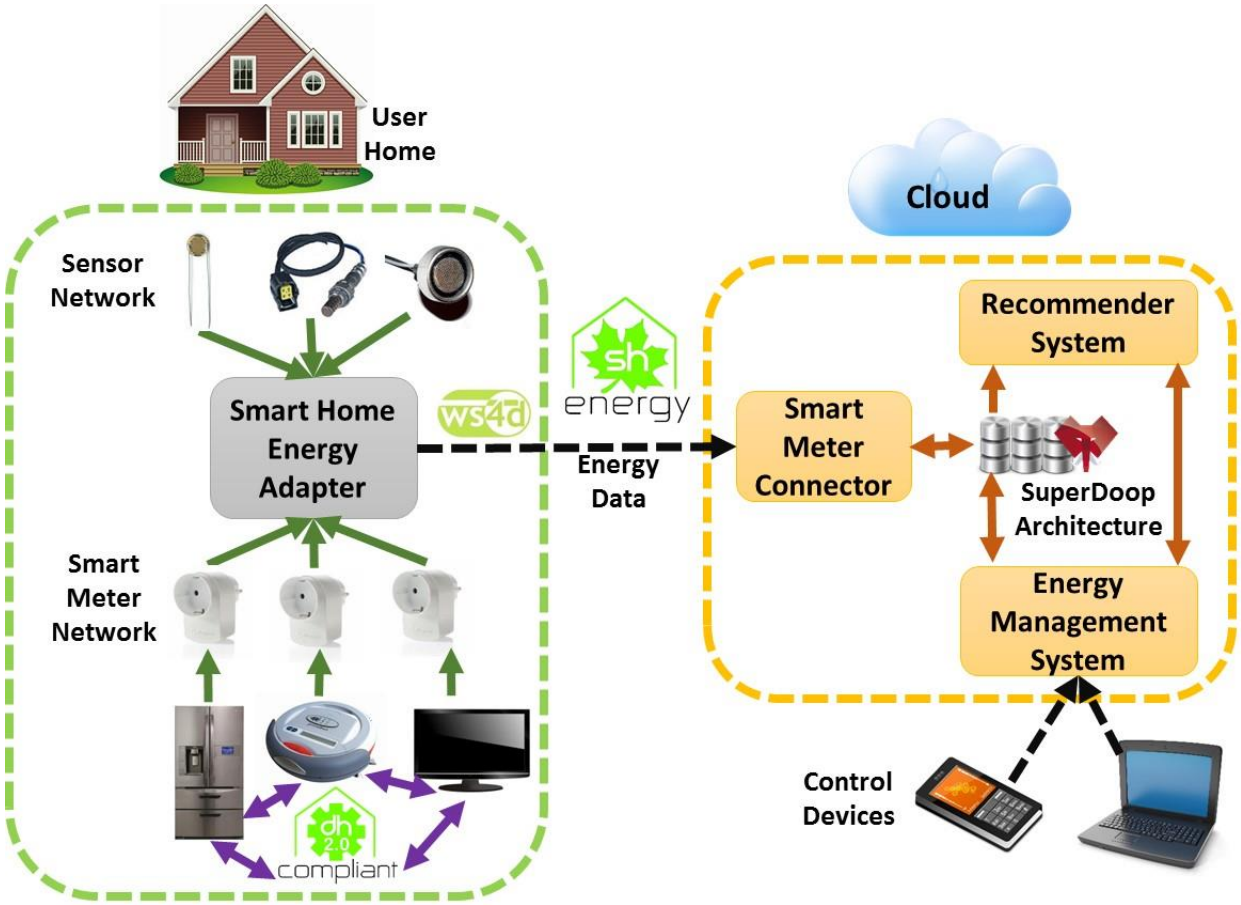


Figure 1: Smart Home Energy architecture schema

- The user home environment, where all the appliances and service robots are able to interoperate among them by means of the DHC communication protocol.
  Specifically, Smart Metering devices and sensors are responsible of acquiring the energy-related home information such as the power consumption, the luminosity or the temperature. Then, the software Smart Home Energy Adapter (SHE Adapter) obtains the energy data from the mentioned devices and sends it to the cloud in a predefined rate.
- The cloud [13] infrastructure is based on SuperDoop technology, developed by Ingenia [14]. It stores large amount of data coming from each home. The recommender system works on the data produced from the monitored homes in order to generate predictions and recommendations through Machine Learning methods. Finally, the user can access by Internet to the EMS interface, to manage and control the operations. For instance, the appliances can be remotely turned on and off

These parts – the user home and the cloud environments – are going to be described in detail in the next subsections.

A. User Home Environment

A Smart Meter is an intelligent measuring device capable of measuring, in real time, the power consumption of the appliances connected to them [16]. In addition, the type of meters used in the project allows to turn them on and off remotely.

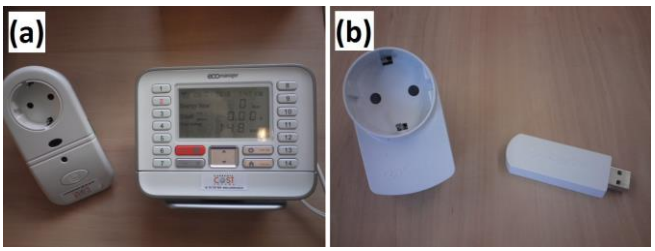


Figure 2: Smart meters (a) EcoManager (Current Cost) and (b) Plugwise

For instance, the plug-based EcoManager by Current Cost [17] and Plugwise [18][19] options shown in Figure 2 (a) and (b) are already integrated in SHE, although the project aims to be independent of the hardware used for obtaining the measures.

Other useful information to define the characteristics of the environment (humidity, number of sunlight hours, rain, wind, etc.) to improve energy efficiency, is collected from the home through a sensor network of different types, such as the ones detailed in Figure 3.

Figure 3: Carbon monoxide (CO), luminosity and temperature sensors

The devices integrating the Digital Home are interconnected through DHC communication protocol. As they come from different manufacturers, it is necessary the implementation of a specific DHC Adapter for each one. A DHC adapter must include the services that are set out below:

- DHC-Security & Privacy service: prevention of fraudulent use, control of the devices, and also, data access by unauthorized agents.
- DHC-Groups service: coordination of collaborative tasks done by different devices belonging to the system.
- DHC-Localization service: Supplying information to locate devices within the home and help them in navigation.
- DHC-Intelligence service: The intelligence is given to the system through the management of rules that control the tasks and the Machine Learning predictions and pattern recognition.
- DHC-Energy service: Incorporation of energy and Smart Grid [20] concepts to the DHC protocol, by means of allowing the user to know the energy consumption data, analyzing the obtained information to improve the energy efficiency of the system and thereby generating savings. Those are the main motivation of SHE project.

In order to clarify the performance role of the DHC-Energy adapter, the main operations are described in detail below.

- Establishment of the charging settings: The types of charging modes (maximum consumption, cost per kWh, etc.) are shown to the customer, which will select the best rate. For instance, the possibility of choosing when a device will have its loading period depending on the tasks that have to be performed, time of the day and the tariffs to be applied. Figure 4 shows a sequence diagram describing the process.

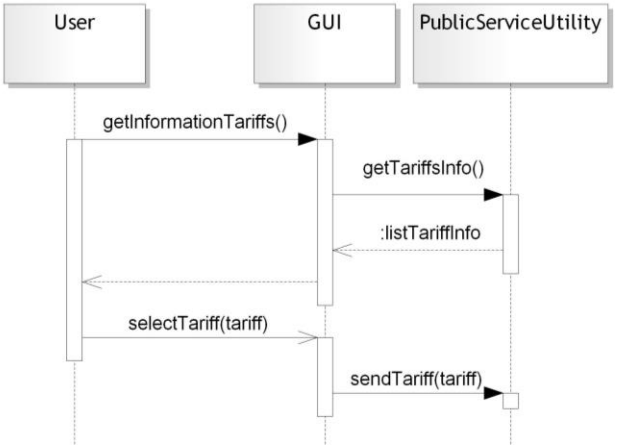


Figure 4: Sequence diagram: Obtain tariff information.

- Device status: It is also important to know the preferences of the user, and compare it with the data that shows the state of the device and the power source (either renewable or normal electrical supply). The process carried out can be seen in Figure 5.

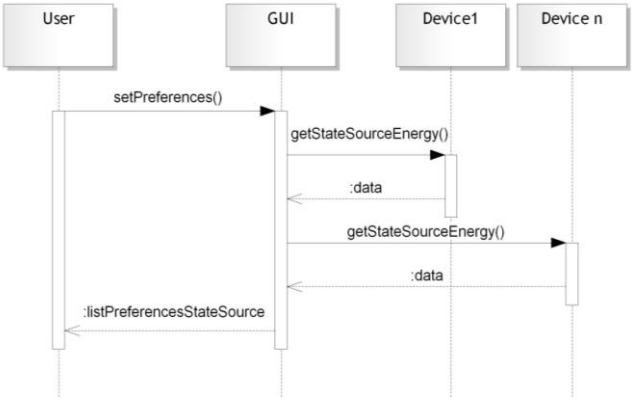


Figure 5: Sequence diagram: Device status

- Energy profiles: The system allows a set of possible energy consumption configurations. When the user does not need the device, the power consumption should be zero ("Off" profile) or minimum in case of awaiting orders ("Stand by" profile). In the operation mode, some devices can be fixed to the highest energy saving rate ("Low" profile) or to lowest energy saving rate, which operates on a full capacity ("High" profile). Some indispensable devices can be set to an especial level without caring the savings ("Emergency" profile).

Moreover, if a device remains in an inactive state, the device must reduce its energy profile to a lower level. The way in which this operation is done is shown in Figure 6.

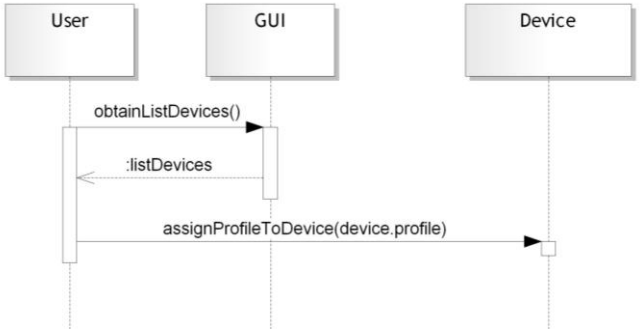


Figure 6: Sequence diagram: Election of an energy profile

For achieving an optimal performance of the system, it is necessary to process the large amount of data generated by the different devices in a fast rate. Therefore, Web Service for Devices (WS4D) technology [21] has been chosen. The main driven behind this selection is that DPWS maintains the philosophy of Service Oriented Architecture (SOA) combined with the convenience of Web Services. This solution means that a Digital Home can transmit the captured information through the DHC adapter to the Cloud.

B. Cloud

It is necessary to store the information from the Digital Home in a centralized way in order to have a database that allows comparative and heuristics analysis. From the user's perspective, the stored information needs to be accessed from any device, anywhere [22].

SHE project uses SuperDoop technology [23], which is a Hadoop [24] and Storm [25] based Open Source Big Data Stack integration, that scales horizontally to solve challenges as acquisition, storage, searching, sharing, analysis and visualization of large data sets on a tolerable response time-frame. This can be used for general-purpose applications, such as communications, banking, security, smart cities, energy efficiency, emergencies, social networks and many others areas.

SuperDoop fits into Lambda Architecture concepts [26]. The designed architecture, which is schematized in Figure 7, has the following main components:

- The energy measurement reception from the environment can be made over different types of networks (wired and wireless). Representational State Transfer Application Programming Interface (REST) API [27] is used for the communication between the Digital Home nodes and the Cloud. It defines an interface among the software components, where an URL represents a resource whose content can be accessed via HTTP protocol via. The use of this technology brings some advantages as portability between different languages, performance and scalability.
- Acquisition Layer is an additional layer to solve the acquisition of the data used by the Speed and Bath layer, described below. This concept has been implemented as a prototype in a flexible way for SHE project using REST and JSON APIs, a temporal storage or cache using MongoDB, and a Redis based system for the speed data layer data insertion.
- The Speed Layer was needed to provide a real time monitoring and control services to users through a Graphical User Interface. This layer needs to aggregate the data, using typical functions as average and summarization for each user, group of devices, and time intervals, in a continuous computing close real time. Finally, this layer provides the information to the Service Layer to make it available to the interface.
- The Batch Layer includes several components to store and process the data, applying data mining and Machine Learning algorithms to acquire a customized knowledge pool for home energy consumption. These algorithms can be executed massively for each home or user, many times a day, to learn and predict the consumption using a large set of data for each execution.
- The Service Layer is the closest to the user, and includes data publishing through a REST API for both layers, the Batch Layer and the Speed Layer, allowing the interface to query, receive and show the data.

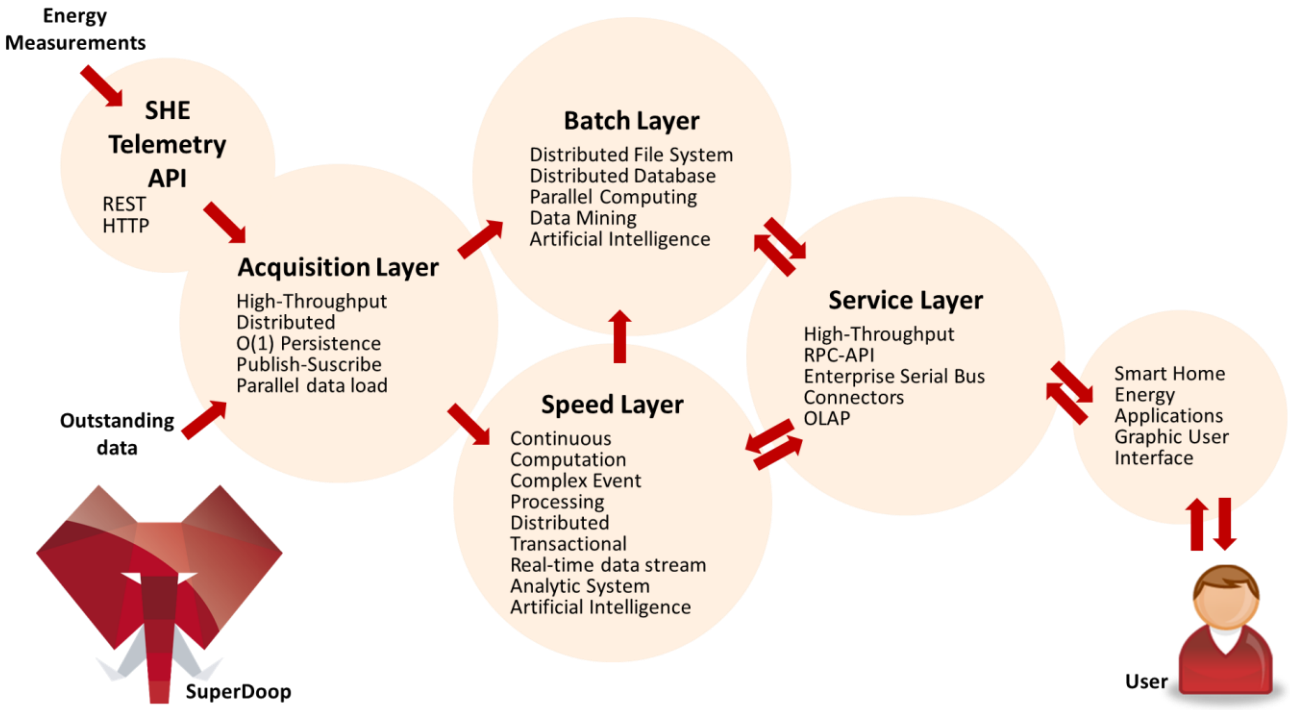


Figure 7: SuperDoop Architecture applied to SHE project

The smart meter connector acts as a middleware between the SHE Adapter, which is in execution in the user home and the SuperDoop platform, in the cloud.

The stored information is used by the system to generate specific recommendations, which will depend on its environmental conditions. This is done through an expert system that, based on its experience, uses rules to model the system. It integrates a knowledge database of the consumption data and a rule editor, which enables the customer to test and simulate the rules. The rule execution is done with an inference engine based on an execution of rules and tree forward.

Going a step further, this element can also generate collaborative recommendations based on actions performed in homes belonging to a similar environment profile. For making this, it would be necessary to define the way in which the distance between two homes or two users is calculated. Taking into account that the users can agree (value 0) or not (value 1) with the completion of an action, the most appropriate algorithms are Tanimoto [28]. The distances between all the users are calculated and stored in a matrix. When a recommendation is required to be given to the user, the algorithm returns the actions that most similar users have done, and that the user has not carried out yet.

It is also possible to extract consumption patterns and thus allow making predictions to anticipate and adapt to other cheaper options.

The user can access remotely and from different devices to the Energy Management System. Through its interface, the consumer is able to manage, control and plan the bill. That is to say, the user has a Smart Billing. Furthermore, in the user home page, the mentioned energy efficient recommendations are offered associated with actions to carry them out.

A complementary useful tool is the incorporation of interactive and customizable graphs to show the information to the user, who is able to manage the energy consumption and consequently improve the energy efficiency.

As it is shown in Figure 8, the consumption insights screen displays the consumption of each day, its distribution in the different hourly periods, the accumulated per month and the distributed consumption among the various days of the week, within the filtering dates, in an interactive and visual way.

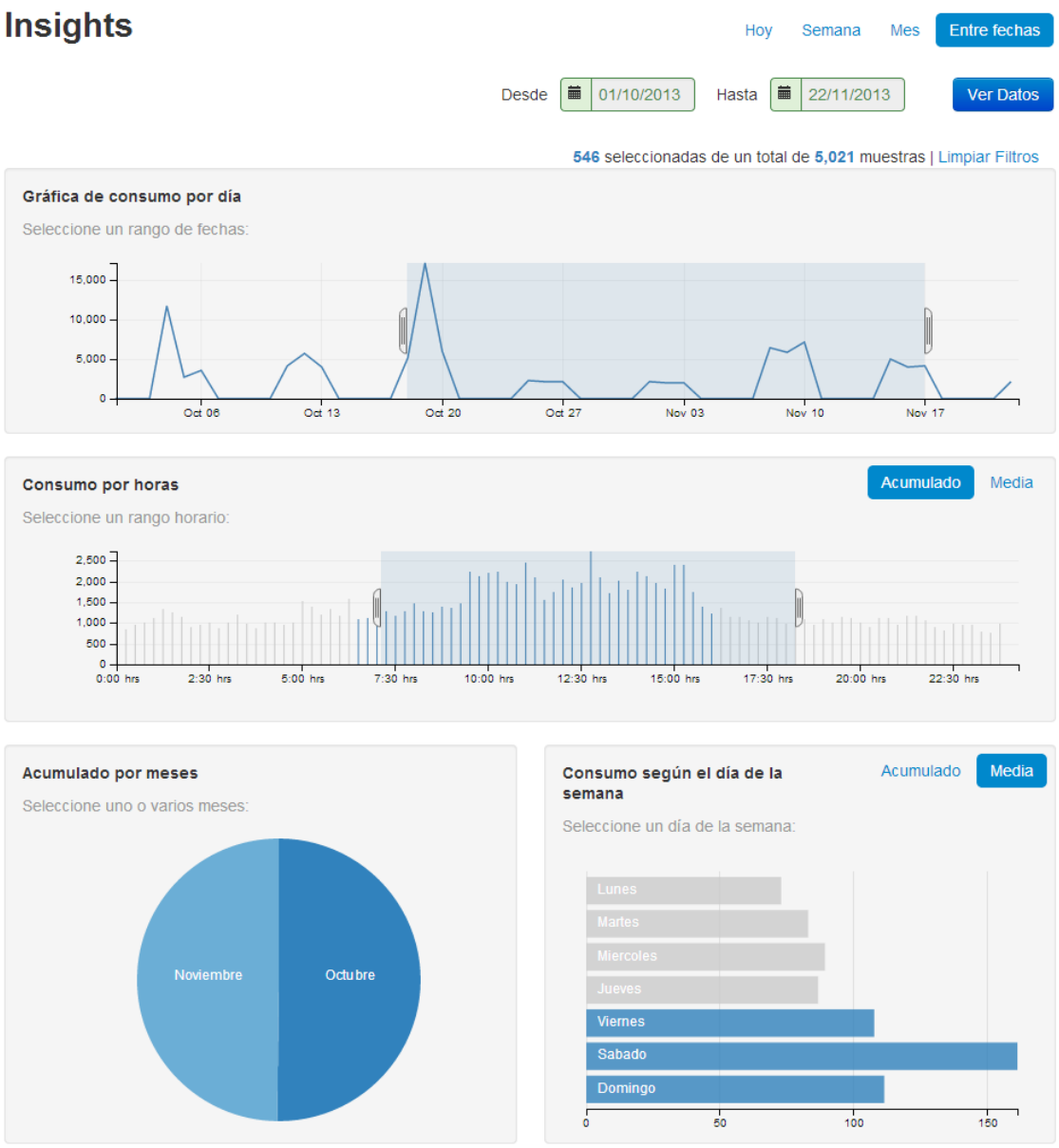


Figure 8: Interface of SIGE: Consumption Insights

Other historical information is presented in different parts of the interface to analyze it and make comparisons. Visual Mining tools are applied to help the user to understand where power is consumed, where it is wasted, etc. In Figure. 9 the weekly user consumption is represented in a heatmap to facilitate the discovering of a behavior pattern visually.

The interface takes into account different aspects of usability, accessibility and, of course, the functional aspects of providing the user with the information that allows monitoring and controlling the appliances presented in the Digital Home.

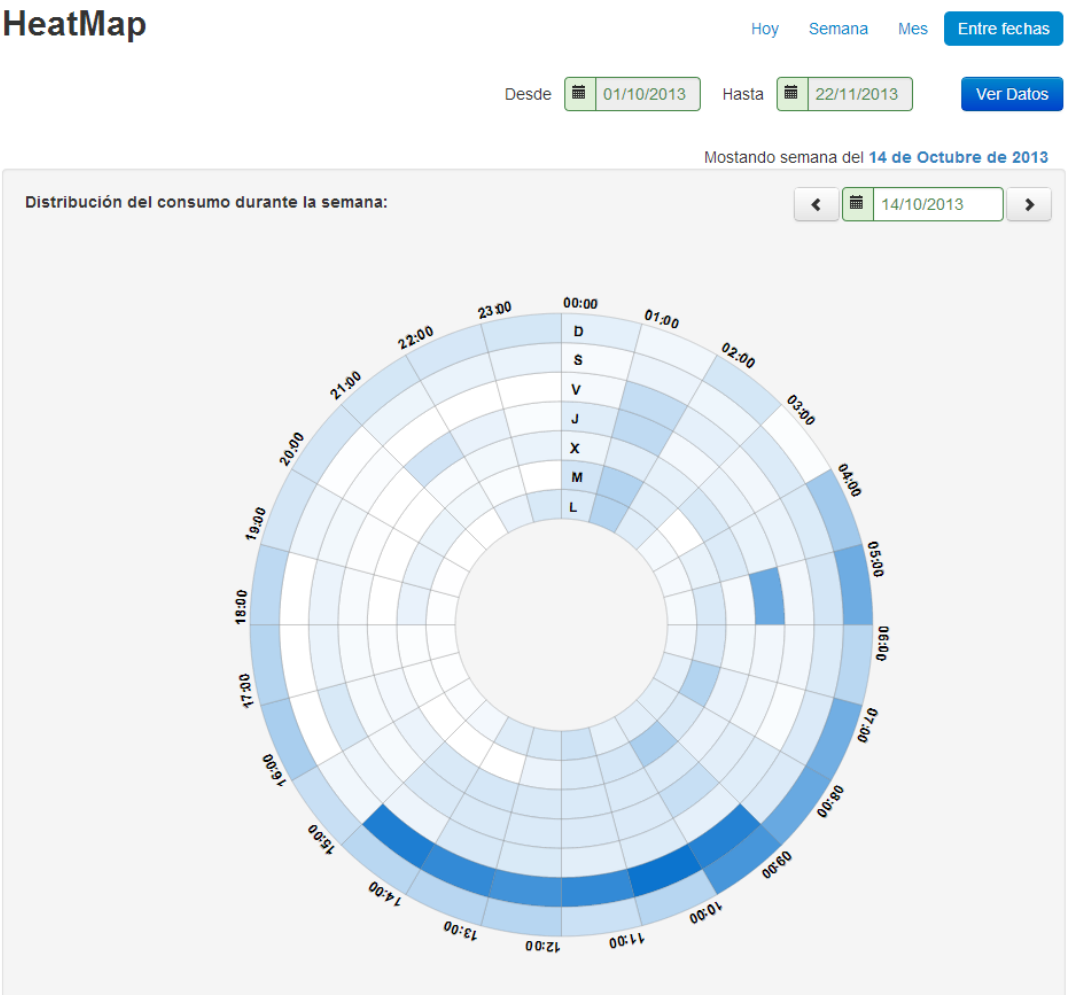


Figure 9: Interface of SIGE: Heatmap

IV. TEST RESULTS

After different attempts to match simulated energy demand with multivariate regression models, the next step was to prove implementation of Artificial Neural Network (ANN) models as suggested in [29]. Several studies about prediction of building energy consumption and temperature forecaster [30][31][32] are based on ANNs and demonstrating a high accuracy in predicting energy demand and consumption in buildings. The main drawback is the need of a learning process for identifying the particular patterns, which define the system performance or behaviour. In this case, a simple ANN (3 hidden layers, using a symmetrical sigmoid as activation function) has been tested with the back-propagation algorithm [33], using as input parameters outside temperature, outside humidity ratio, previous hourly energy demand, among others. The obtained results have satisfied survey expectation since using short-term ANN for energy demand prediction has shown an average error of 3% during a complete year simulation what constitutes insignificant deviation compared with typical estimations that usually energy experts make in energy audits (more than 20%). The comparison between the energy demand (heating and cooling) of a small building simulated by EnergyPlus ® and the ANN's outputs is shown in Figure 10.

The test was performed with no seasonal discrimination, which demonstrates that the results could actually be reasonable to predict easily, without high accuracy, the energy demand and thus, the energy consumption, in case of we know the rated efficiency of domestic facilities.

Even, the mentioned accuracy has been achieved with simple ANN composed by 5-neuron layers designed for estimating the thermal energy demand of a house as output, paving the way to design a robust energy forecaster. In order to include the data in the forecaster, the value of input parameters have to be established between 0 and 1 (dimensionless).

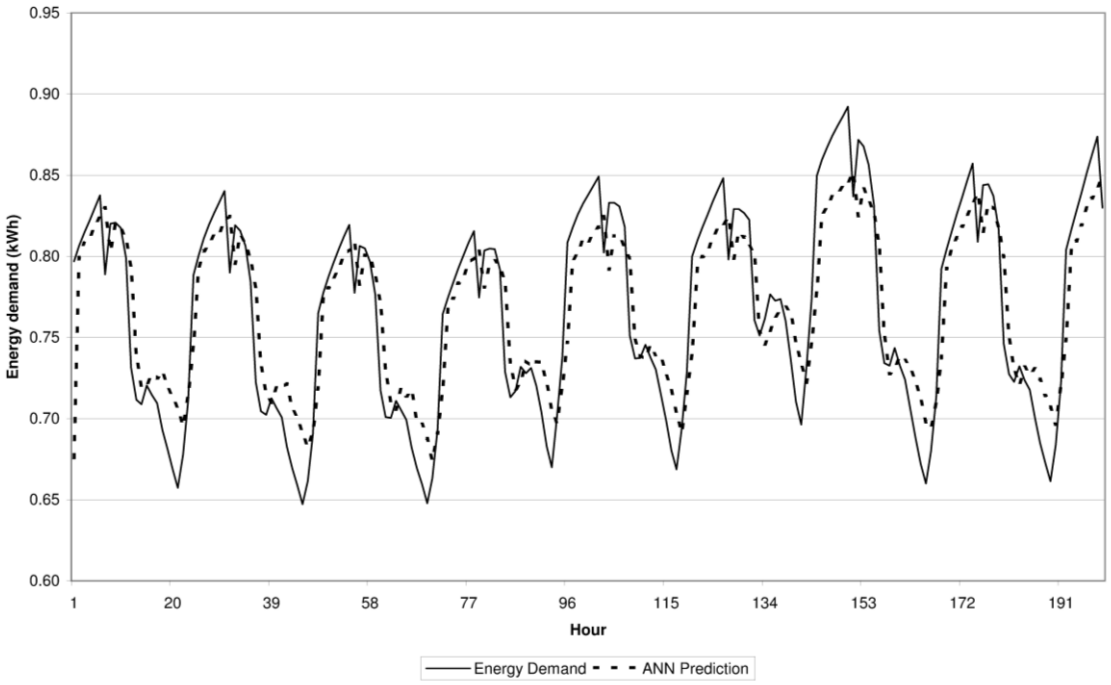


Figure 10: Comparison building energy demand and ANN prediction

For instance, in this study, the outside temperature T_{out}, humidity ratio φ and previous energy demand d_{k-1} have been transformed according with following equations (1), (2) and (3):

$$T_{out}^{*} = \frac{T_{out} - T_{\min}}{T_{\max} - T_{\min}} \quad (1)$$

$$\phi^{*} = \frac{\phi}{100} \quad (2)$$

$$d_{k-1}^{*} = \frac{d_{k-1} - d_{\min}}{d_{\max} - d_{\min}}, \quad (3)$$

Min and max subscripts are related to the minimum and maximum historical temperature and potential demand respectively. When ANN calculate the current energy demand d_k^*, the inverse transformation is easy to understand according to (3). As mentioned above, ANNs need a learning process to modify the synaptic weights that allow the network to integrate the performance of a complex system.

The current assessment did not consider the different seasons, months, holidays, etc. in order to obtain greater precision since another most sophisticated algorithm called support vector machine (SVM) considers the identification of behavior patterns, related to the likelihood of an event occurs based on input parameters [34]. This mixture of artificial intelligent and statistical approach, e.g., regression models, confers a highly effectiveness in solving non-linear problems, with reduced training data, as our current project, what shows this model as the ideal candidate for energy consumption prediction in buildings. In addition, due to the robustness of this prediction method, potential perturbations, which could appear by adapting the smart home system to new buildings, will be mitigated as it is mentioned in [35].

V. CONCLUSION AND FUTURE WORK

The system described in this paper allows to determine that an open stage of interaction between devices and the Smart Grid can be set by providing more capabilities than pure traditional energy efficiency (such as accounting and reductions in consumption). It also permits the establishment of a consumption profile for the different heterogeneous appliances, which a customer has at home, as well as a referral system in the Cloud associated with business intelligence that allows reducing even more the energy expenditure. All this is done in a distributed way but through a single point where the user interacts. Decisions could be

ineffective or even counterproductive if a holistic approach would not have considered.

Cloud technology offers an elastic and resilient solution without requiring a high-capacity storage infrastructure at household level. It also facilitates the management and maintenance of the integrity, security and availability of data. In addition, this solution provides facilities for transparent software updating, because most of the software is centralized and non-distributed on each node.

Specifically, the application of SuperDoop to the metering data allows doing both a batch and a continuous real-time processing of the measurements, working with a large set of data taken along the time from a large set of homes and historical database. It facilitates the generation of configurable and customizable reports, and recommendations, among other functions and independent of the measuring device.

The proposed solution applies ANNs and Machine Learning algorithms using stored and real-time data, thus can be used to acquire personalized consumption knowledge for each home. At the beginning of the learning, real data is temporally replaced with simulated datasets of homes and historical weather data and climate zones, so as the system does not have historical data yet. The prebuilt knowledge can be applied to a classification algorithm, until the system have enough real data, a real customized knowledge progressively learns the real behavior and forgets the knowledge based on simulated data. Thus, the system would acquire the ability of predicting consumption for each home through the customized home energy knowledge.

Summarizing, this technology has advantages over other approaches, because it is open, distributed, and scalable. Besides, it requires little or no configuration by the end user. The use of Big Data techniques does not require initial investment. This method and technology can also be used for other stakeholder solutions that are beyond the scope of this communication.

ACKNOWLEDGMENT

This project is funded by the Ministry of Economy and Competitiveness of Spain (IPT-2011-1237-920000) and FEDER funds

We are also very grateful to all the members of the consortium (Ingenia, Satec, Ingho, Tecopysa, Cotesa, IAT, and University of Oviedo) and to the programmers of the entities participating in the project (Sergio Tudela, José Farfán, Consuelo Extremera, Felipe Fresneda, Sergio Rodríguez, Manuel Rodríguez, José Antonio Luque, Francisco Jurado, Víctor García and José María Ocón).

REFERENCES

[1] I. González, M. Rodríguez Fernández, J. J. Peralta, and A. Cortés, "A Holistic Approach to Energy Efficiency Management Systems," in *ICSEA 2012, The Seventh International Conference on Software Engineering Advances*, 2012, pp. 415–420.

[2] I. González Alonso, O. Álvarez Fres, A. Alonso Fernández, P. G. del Torno, J. M. Maestre, and M. Almudena García Fuente, "Towards a new open communication standard between homes and service robots, the DHCompliant case," *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 889–900, 2012.

[3] R. Spinar, P. Muthukumaran, R. de Paz, D. Pesch, W. Song, S. A. Chaudhry, C. J. Sreenan, E. Jafer, B. O'Flynn, and J. O'Donnell, "Efficient building management with IP-based wireless sensor network," in *EWSN 2009, 6th European Conference on Wireless Sensor Networks*, 2009.

[4] I. Lampropoulos, P. P. van den Bosch, and W. L. Kling, "A predictive control scheme for automated demand response mechanisms," in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, 2012, pp. 1–8.

[5] Universal Consortium, "UniversAAL Website," 2013. [Online]. Available: http://universaal.org/index.php/en/. [Accessed: 25-Nov-2013].

[6] Infobótica Research Group, "DH Compliant Website," 2013. [Online]. Available: http://156.35.46.38/dhc2/. [Accessed: 25-Nov-2013].

[7] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring behavior in home using a smart fall sensor and position sensors," in *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On.*, 2000, pp. 607–610.

[8] P. Dobrev, D. Famolari, C. Kurzke, and B. A. Miller, "Device and service discovery in home networks with OSGi," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 86–92, 2002.

[9] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, "Monitoring behavior in home using a smart fall sensor and position sensors," in *Microtechnologies in Medicine and Biology, 1st Annual International, Conference On. 2000*, 2000, pp. 607–610.

[10] I. O. Media, *Big Data Now: 2012 Edition*. O'Reilly Media, 2012.

[11] Y. Ding, N. Namatame, T. Riedel, T. Miyaki, and M. Budde, "SmartTecO: Context-based Ambient Sensing and Monitoring for Optimizing Energy Consumption," in *Proceedings of the 8th ACM International Conference on Autonomic Computing*, 2011, pp. 169–170.

[12] Infobótica Research Group, "Smart Home Energy Project Websiste," 2013. [Online]. Available: http://156.35.46.38/she/. [Accessed: 25-Nov-2013].

[13] J. Rhoton and R. Haukioja, *Cloud Computing Architected: Solution Design Handbook*. Recursive Press, 2011.

[14] Carlos Bentabol, "El concepto Smart City," *Revista Ingenia*, vol. 57, 2013.

[15] Ingeniería e Integración Avanzadas S.A. (Ingenia), "IT solutions for Smart Cities," 2011. [Online]. Available:

http://www.ingenia.es/en/productos_servicios/it-solutions-smart-cities. [Accessed: 26-Jun-2013].

[16] F. Casellas, G. Velasco, F. Guinjoan, and R. Piqué, "El concepto de Smart Metering en el nuevo escenario de distribución eléctrica," *Actas del Seminario Anual de Automática y Electrónica Industrial, SAAEI*, 2010.

[17] Current Cost, "Reducing your energy bills so you can live a greener life," 2012. [Online]. Available: http://www.currentcost.com/. [Accessed: 12-Jul-2013].

[18] "Plugwise Website," 2013. [Online]. Available: http://www.plugwise.com/. [Accessed: 25-Nov-2013].

[19] Plugwise, "The future of energy monitoring in today's buildings," *NRG Magazine Edition*, 2013.

[20] S. Chen, S. Song, L. Li, and J. Shen, "Survey on smart grid technology," *Power System Technology*, vol. 33, no. 8, pp. 1–7, 2009.

[21] Golatowski, Frank, Bobek, Andreas, and Zeeb, Elmar, "Web Services for Devices (WS4D) Website," 2012. [Online]. Available: http://ws4d.e-technik.uni-rostock.de. [Accessed: 14-Dec-2012].

[22] L. Spaanenburg and H. Spaanenburg, *Cloud connectivity and embedded sensory systems*. Springer, 2010.

[23] Ingenia S.A., "Ingeniería e Integración Avanzadas." [Online]. Available: http://www.ingenia.es/es. [Accessed: 14-Dec-2013].

[24] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, 2010, pp. 1–10.

[25] V. V. Khadilkar, M. Kantarcioglu, and B. Thuraisingham, "StormRider: harnessing storm for social networks," in *Proceedings of the 21st international conference companion on World Wide Web*, 2012, pp. 543–544.

[26] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications, 2013.

[27] M. Masse, *REST API Design Rulebook*. O'Reilly Media, 2011.

[28] C. Cechinel, M.-Á. Sicilia, S. Sánchez-Alonso, and E. García-Barriocanal, "Evaluating collaborative filtering recommendations inside large learning object repositories," *Information Processing & Management*, vol. 49, no. 1, pp. 34–50, Jan. 2013.

[29] R. Parsons, "ASHRAE Handbook Fundamentals," *American Society of Heating, Refrigerating and Air-conditioning Engineers*, 1997.

[30] P. A. González Lanza and J. M. Zamarreño Cosme, "A short-term temperature forecaster based on a state space neural network," *Engineering Applications of Artificial Intelligence*, vol. 15, no. 5, pp. 459–464, 2002.

[31] P. A. González and J. M. Zamarreno, "Prediction of hourly energy consumption in buildings based on a feedback artificial neural network," *Energy and Buildings*, vol. 37, no. 6, pp. 595–601, 2005.

[32] B. B. Ekici and U. T. Aksoy, "Prediction of building energy consumption by using artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, 2009.

[33] R. Rojas, *Neural Networks: A Systematic Introduction*, 1st ed. Springer, 1996.

[34] R. E. Edwards, J. New, and L. E. Parker, "Predicting future hourly residential electrical consumption: A machine learning case study," *Energy and Buildings*, vol. 49, pp. 591–603, 2012.

[35] H. Zhao and F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

Generation and Assessment of Urban Land Cover Maps Using High-Resolution Multispectral Aerial Cameras

Joachim Höhle
Department of Planning, Aalborg University
Aalborg, Denmark
Email: jh@land.aau.dk

Michael Höhle
Department of Mathematics, Stockholm University
Stockholm, Sweden
Email: hoehle@math.su.se

***Abstract*—New aerial cameras and new advanced geo-processing tools improve the generation of urban land cover maps. Elevations can be derived from stereo pairs with high density, positional accuracy, and efficiency. The combination of multispectral high-resolution imagery and high-density elevations enable a unique method for the automatic generation of urban land cover maps. In the present paper, imagery of a new medium-format aerial camera and advanced geoprocessing software are applied to derive normalized digital surface models and vegetation maps. These two intermediate products then become input to a tree structured classifier, which automatically derives land cover maps in 2D or 3D. We investigate the thematic accuracy of the produced land cover map by a class-wise stratified design and provide a method for deriving necessary sample sizes. Corresponding survey adjusted accuracy measures and their associated confidence intervals are used to adequately reflect uncertainty in the assessment based on the chosen sample size. Proof of concept for the method is given for an urban area in Switzerland. Here, the produced land cover map with six classes (building, wall and carport, road and parking lot, hedge and bush, grass) has an overall accuracy of 86% (95% confidence interval: 83-88%) and a kappa coefficient of 0.82 (95% confidence interval: 0.78-0.85). The classification of buildings is correct with 99% and of road and parking lot with 95%. To possibly improve the classification further, classification tree learning based on recursive partitioning is investigated. We conclude that the open source software "R" provides all the tools needed for performing statistical prudent classification and accuracy evaluations of urban land cover maps.**

Keywords-land cover map; classification; assessment; thematic accuracy; multispectral camera; map revision

I. INTRODUCTION

Land cover maps are an important product of the mapping and GIS industry. The coverage of the landscape with vegetation, soil or man-made constructions may be automatically mapped and analysed. Images taken from an airplane or a satellite are used to derive land cover maps automatically. The intensity values of picture elements (pixels) in different bands of multispectral images are used to determine the different types of objects. It is difficult to distinguish between objects on the ground and objects above the ground, e.g., between buildings and parking lots or trees and hedges. Land cover maps can be generated with a higher thematic accuracy when information on elevations is included [1]. Today, advanced processing tools are able to generate elevations of high accuracy and density from imagery [2]. The generation of land cover maps and their applications will benefit from these innovations.

Of special interest are built-up areas where small objects have to be mapped and where changes over time frequently occur. For such areas, land cover maps have to be produced by means of high-resolution images. Their ground sampling distance (GSD) is a few centimeters. The use of a new high-resolution multispectral aerial camera will be a characteristic of this contribution. In addition, the applied methodology uses a new approach in mapping. The images are used for the generation of a point cloud of high density and of regular structure (grid). Besides the spatial coordinates of a geodetic reference system and a map projection (e.g., easting, northing, and elevation) the individual points may also have attributes such as "height above ground" and "vegetation index". The land cover map is generated by a classification procedure and the result is then visualized by simple plot commands. In order to assess the thematic accuracy of the land cover maps, one typically selects a sample of points in the map, derives their true classes using manual procedures and then extrapolates the accuracy results to the entire map using statistical principles [3].

The outline of this paper is as follows: After the introduction (Section I) the state of the art in the generation of land cover maps is discussed and our aims are formulated (Section II). The new generation of aerial cameras and processing tools is presented in Sections III and IV. Our method for the generation of urban land cover maps is explained in Section V. Special attention is given to the accuracy assessment of these land cover maps (Section VI). In Section VII our method is applied to a test area in Switzerland. The obtained results on the test area are evaluated in Sections VIII and IX. A conclusion on the work carried out and suggestions for future work end the paper (Section X).

II. LAND COVER MAPS AND THEIR APPLICATIONS

Land cover maps have a number of different classes (categories), which are visualized in a two-dimensional (2D) map by different colours. The standard methods classify the images by means of training areas where the truth (reference) is known from field work or manual interpretation of images. The automatic classification of the images uses the intensity values of image elements (pixels) in different bands of the

images and applies various statistical methods. More advanced methods form larger areas (objects) first and classify them in a second step. This object-based classification uses a number of attributes like shape, size, and texture and generates land cover. The land cover map has to be georeferenced. This means that the orientation of the images and a digital terrain model (DTM) have to be known in advance. The classification may use the original images or orthoimages - in the first case the classification result has to be rectified and orthogonalized. The derived land cover map is assessed by a few independent check points or check areas. The results are presented by means of an error (confusion) matrix.

The usage of elevations as an attribute of objects has been tried before. In [1], a solution for automatic generation of land cover maps has been proposed and applied to a one hectare large test area. Elevations, heights, and vegetation indices have been derived from images only. True-colour orthoimages and false-colour stereo pairs were used for the determination of the reference. The overall accuracy of the derived land cover map with four classes (building, road and parking lot, tree and hedge) has been 79%. The use of stereo-vision was the preferred approach in the assessment of the thematic accuracy.

In [4], elevations were derived from laser scanning and have been used together with aerial images and cadastre maps for the generation of a land cover map. Ten urban and seven agriculture classes have been determined for the city of Valencia/Spain. The overall accuracy of the derived land cover map was 85%. Reference [5] describes an object-oriented classification where 81% of all residential buildings are correctly determined for a test area located in Austin, Texas/U.S.A. Attributes of objects are derived from laser scanning, aerial images, and road maps.

Machine learning algorithms are applied for the urban land cover classification in [6]. The classification can either be based on pixels or object features. A combination of pixels, object features (area, length, etc.), and layer values (mean, standard deviation, etc.) has been suggested in Reference [7].

A land cover map has many applications and can be used for all types of planning. View shed analysis and studies of propagation of noise or electromagnetic waves are applications in engineering. An important application is the revision of topographic data bases. This task requires a high positional and thematic accuracy. In order to detect changes and errors in topographic databases the generation of a land cover map may be the first step in the revision process. The superimposition of the land cover map with the topographic database will indicate which objects of the topographic database have to be added, deleted, or changed in size or shape. Recent studies in revision of topographic databases are carried out in [8] [9]. The exact actions depend on what types of data have to be revised. There are topographic databases in 2D, 2.5D, and 3D. Sometimes, the important objects are updated only, e.g., buildings, roads, and trees. Land cover maps are also used for studies in town development. The changes in areas of buildings, traffic, and vegetation over several years are studied in [10]. In that investigation, the land cover maps are derived from vector maps and low-resolution satellite images. The applied classification of the images uses intensity values of pixels only.

Thus, our aims for this paper are:
1) Generation of urban land cover maps by applying decision trees based on height and vegetation information derived from a new type of aerial images. 2) Assessing the thematic accuracy using overall and class specific accuracy measures. 3) To do so by a class-wise stratified sampling design of reference points, which allows for precise estimation of per class accuracy quantities. 4) To illustrate that all the necessary computation can be done by "R" and its extension packages.

In comparison to the investigation in [1] we now focus here on the accuracy assessment as point estimates and confidence intervals in the light of a class-stratified sampling design. Furthermore, classification is now improved using additional classes, and a comparison is made with machine learning derived decision trees. Finally, all computations in the proof-of-example section are now based on a larger test area than previously.

III. NEW DIGITAL AERIAL CAMERAS

With the development of new advanced digital aerial cameras, the generation of land cover maps has new tools at its disposal. There are different types and models of aerial cameras. Three of the most advanced cameras will be discussed in the following: the Hexagon/Intergraph DMCII 250, the Microsoft UltraCam Eagle, and the Hexagon/Leica Geosystems RCD30 camera. Details of the cameras are described in publications [11] [12] [13]. All three cameras are frame cameras, which can produce images of high resolution and high geometric quality. They are designed for mapping tasks. The produced images have different features; the major ones are listed in Table I.

There is a considerable difference in the format of the output image. The RCD30 is a medium-format camera; the other two cameras are considered as large-format cameras. A larger format requires less flying time in order to cover an area to be mapped assuming that the images of all cameras have the same GSD. The necessary length of the flight is an economic factor, and large-format cameras have an advantage in projects covering large areas. All three cameras can produce black & white, colour, and false-colour images simultaneously. Newly designed lenses match the resolution of the sensors and make high image quality possible. The

TABLE I. FEATURES OF THREE NEW DIGITAL AERIAL CAMERAS

Features		***UC Eagle***	***DMCII 250***	***RCD30***
pixel size	[µm]	5.2	5.6	6.0
focal length	[mm]	80	112	50, 80
image size (in flight direction)	[pel]	13 080	14 656	6708
	[mm]	68.0	82.1	40.2
image size (across flight direction)	[pel]	20010	17216	8956
	[mm]	104.1	96.4	53.7
number of pixels per image	[MP]	262	252	60

cameras are calibrated, and the obtained calibration data are used to correct the images in geometry and radiometry. Additional sensors for position (GNSS) and attitude can be supplemented and will support accurate georeferencing of the imagery. More details have to be mentioned on the RCD30 camera because its images will be used in the following examples. The colour images are produced from one charge-coupled device (CCD) with Bayer filters. The infra-red band is imaged by a second CCD of the same high resolution (pixel size=6 µm x 6 µm). Image motion is compensated mechanically in two axes. The output images are corrected for distortion, light-fall off of the lens, and non-uniformity for dark signals. Two different lenses can be used without the need of re-calibration. In order to obtain images with GSD=0.05 m, they have to be taken from 417 m above ground with a 50 mm lens. One frame will cover 0.15 km^2 on the ground. A gyro-stabilized mount can be used which will prevent big tilts of the imagery.

IV. NEW PROCESSING TOOLS

The generation of land cover maps from images requires the use of specific software tools. They span from general image processing to dedicated software for photogrammetry, remote sensing, and geographic information systems (GIS). The characteristic of our approach is the use of digital elevation models, which are derived from high-resolution images. The latest developments in this field may enhance the performance of the previously suggested approach of the generation of land cover maps [1]. One of the problems to overcome is the large amount of data, which might make it necessary to divide the work into smaller units. One 64-bit computer with 8 GB RAM (as is at disposal in this investigation) is not an optimal processor for this task. More computer power is necessary to solve the task for large areas. The many different software tools are expensive if they have to be acquired from commercial software vendors. However, an increasing amount of freeware and open source tools has become available to conduct some of the tasks. Such tools are preferred whenever possible. Specifically, we use the open-source software environment "R" [14] for statistical computing and visualization and, hence, one aim of the present paper is to demonstrate its use in the generation and assessment of land cover maps.

A. Programs for the generation of digital elevation models

Digital elevation models can automatically be derived from two or more images covering the same area on the ground. Corresponding image parts are matched using the intensity values of image elements (pixels). Spatial ground coordinates are intersected by means of the corresponding pixels in two or more georeferenced images. The resulting point cloud can be transferred into a regular grid of points representing a digital surface model (DSM). Such software programs are pretty complex and their development requires many man years. Professional programs are, therefore, expensive. Many parameters have to be tuned by the user in order to derive elevations of high accuracy and completeness, which in turn means that some expertise is necessary in order to derive good results.

The problems to be overcome are the huge amount of data and the mismatches due to the lack of contrast and structure in the images. International tests reveal elevation accuracies of $RMSE_Z$=4cm at large-scale images with GSD=8 cm [15]. The density of the point cloud may be extremely high, e.g., 156 points/m^2 at GSD=8 cm [16]. This means that the derived DSM may have the resolution of the images. The use of distributed processing and dedicated hardware make a high performance in the calculations possible. Reference [17] reports about a benchmark test where 997 million points are derived from 1562 images (filling 1.44 TB) in 2.5 hours or six seconds/image.

The DSM is an intermediate product only. For the generation of land cover maps the heights of the objects (buildings, trees, etc.) are required. This is accomplished by filtering the DSM so that only elevations of the bare earth (terrain) will remain. This process requires editing and checking. From the remaining terrain points a grid of elevations is interpolated, which represents a DTM. The editing software is a complex software tool, which allows for measuring, correcting, and visualizing of 3D data. The difference between DSM and DTM is the normalized DSM (nDSM), which is the required input of the classification program. The nDSM is derived using a C++ program, which performs the matching between the two point clouds by using a linear search to identify the first neighbour within a distance of 0.05 m. As a consequence, the search of corresponding elevations in the two elevation models (DSM and DTM) is quite time-consuming. Improvements could be obtained by exploiting the spatial structure of the data in the search.

B. Programs for the generation and assessment of urban land cover maps

The generation of the land cover map and the assessment of the results are developed using the statistical software environment "R", which is an open source environment for statistical computing and graphics. The programming can be both object-oriented and functional in style. Furthermore, a large number of open source extension packages (created by an active community) provide the newest methodological functionality for many visualization and statistical tasks. For example, the package "survey" [18] can be used for computing accuracy measures in complex survey situations, "rpart" [19] implements recursive partitioning for classification trees, and "vrmlgen" enables the visualization of land cover maps in 3D [20].

V. APPLIED METHOD FOR THE GENERATION OF LAND COVER MAPS

The applied method uses the nDSM derived from images and supplements it with additional attributes. The points of the nDSM are arranged in a regular grid of high density. The spacing between points has been selected as a multiple of the image element (pixel) on the ground. The new unit is called a cell in the 2D space or cube in the 3D space. The attributes to be added characterize the object to be mapped. They are

TABLE II. SPATIAL POINT CLOUD WITH ATTRIBUTES

Easting [m]	*Northing* [m]	*Elevation* [m]	*nDSM* [m]	*NDVI*
537129.2	5228938.6	486.5	0.2	0.31
537129.2	5228938.7	488.5	2.3	0.28
537144.5	5228987.4	486.4	0.1	0.01
537128.0	5228938.3	490.8	4.2	0.07

derived from the images only. The first attribute is the normalized height which is the height of objects above ground (nDSM). The normalized difference vegetation index (NDVI) is the second attribute, which characterizes vegetation. Classes of objects are obtained by defining thresholds for the nDSM and NDVI values. Table II shows an example of such a point cloud with its attributes. A classification program will assign a class to each cell/cube of the spatial point cloud. The classification scheme is depicted in Figure 1. The chosen classes are 'grass', 'road and parking lot', 'tree and hedge', and 'building'. This classified point cloud can then be plotted in 2D or 3D. This simple classification may be improved by selecting more steps in the thresholds and by using information which characterizes a class.

VI. ASSESSMENT OF ACCURACY

The assessment of the accuracy of the generated land cover map is an important task to ensure an appropriate quality of the maps. It concerns the positional and the thematic accuracy. The positional accuracy is checked within the production of the land cover map. It is of great importance that all data are in the same reference system and that positional and thematic accuracies of the reference data are superior to the classified data. The assessment of the thematic accuracy is conducted by means of an error matrix. This measure is widely accepted and described [3] [21]. The employed accuracy measures based on the error matrices as well as the design of the accuracy sample will be summarized in the following sections.

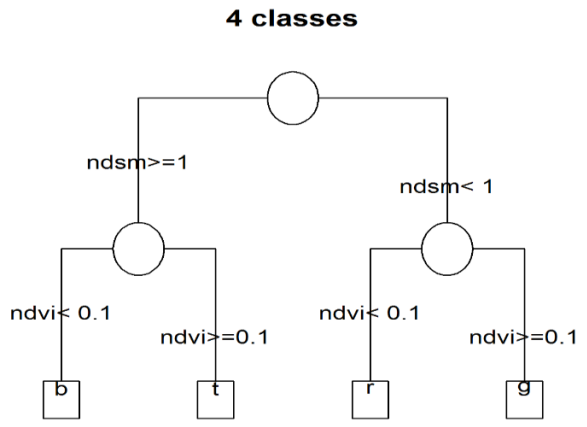


Figure 1. Classification tree for a land cover map with four classes. The abbreviations are: b='building', t='tree and hedge', r='road and parking lot', g='grass'.

A. Accuracy assessment and error matrix

The classified data are compared with reference data. Assume the land cover map consists of a total of N units (cells), which each is classified into one of k classes. Selecting a cell at random let $p_{ij}=P(\text{class}=i, \text{ref}=j)$, $i=1,...,k$, and $j=1,...,k$, be the joint probability that the classifier claims class i for this class while the reference says it is class j. Note that the sum over these $k \cdot k$ probabilities will be 1. The probabilities can be represented by a $k \cdot k$ matrix, where the rows represent the classified data and the columns the reference data. The probabilities in the diagonal of the matrix denote the correctly classified sample units for each of the classes; all probabilities outside the diagonal represent errors.

The sum of the probabilities in the diagonal is the so called "overall accuracy", i.e., it is the probability that a randomly selected unit (cell) will be correctly classified:

$$oacc = \sum_{i=1}^{k} p_{ii}. \qquad (1)$$

Other accuracy measures are the so called class-wise "producer's accuracy" and "user's accuracy" [3]. The "producer's accuracy for class j" is defined as

$$pacc(j) = P(\text{class} = j \mid \text{ref} = j) = \frac{p_{jj}}{p_{+j}} = \frac{p_{jj}}{\sum_{i=1}^{k} p_{ij}}, \qquad (2)$$

and the "user's accuracy for class i" as

$$uacc(i) = P(\text{ref} = i \mid \text{class} = i) = \frac{p_{ii}}{p_{i+}} = \frac{p_{ii}}{\sum_{j=1}^{k} p_{ij}}. \qquad (3)$$

The estimation of these accuracy measures for a specific thematic map depends on an estimation of the p_{ij}'s from an "accuracy" sample of size n from the population. Here, n is typically much smaller than N. In the case where simple random sampling with replacement is used to draw the accuracy, the sample estimates are

$$\hat{p}_{ij} = n_{ij} / n,$$

where n_{ij} is the number of cells in the accuracy sample with classifier class i and reference class j, and

$$n = \sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij}$$

the total size of the accuracy sample. Typically, the entire error matrix is then just represented by the n_{ij} counts in the sample. However, in more complex survey designs, the probabilities have to be estimated according to the sampling design. Hence, any error matrix representing an accuracy

sample has to be accompanied by a statement of the selected sample design in order to facilitate proper estimation.

To assess the thematic accuracy of the land cover map by the above measures, one has to decide on an adequate design and sample size of the accuracy sample. Since one of our aims is to estimate user accuracy for each class up to a given precision we use a stratified simple random (STSI) sampling, selecting n_c samples for each class by simple random sampling (without replacement) and, hence, a total sample size of $n=k \cdot n_c$. In the subsequent analyses of our stratified accuracy sample it is, thus, important to take the stratification into account, because classes might be over- or under-sampled compared to their overall distribution in the map. In this case (and ignoring any finite population correction), we will use

$$\hat{p}_{ij} = \left(N_i / N\right) \cdot \left(n_{ij} / n_{i+}\right)$$

as estimates [27], where N_i is the total number of units in the population, which are of class i and

$$n_{i+} = \sum_{j=1}^{k} n_{ij}$$

is the number of units in the accuracy sample, which have classifier class i. A particular strength of "R" is here the availability of the package "survey" [18], which facilitates the computation of such survey weighted accuracy measures - even in complex survey situations. As an example, the above stratified design is easily specified using the `svydesign` function after which the accuracy measures can be computed using the function `svyciprop`.

B. Reference data

The reference data should be independent, of high accuracy, and reliable. The collection of the reference data has to be carried out nearly at the same point of time as the production of the land cover map, so that a change in the landscape and vegetation will not have an effect on the results of the assessment. The efforts for collecting reference data have to be economically justified, i.e., they should be a fraction of the costs for generating the land cover map.

It may be difficult to fulfil all of these requirements. The land cover map is generated from a vegetation map (derived from false-colour orthoimages based on a DTM) and a normalized DSM. Both digital elevation models are automatically derived from images in true colour. The reference data may be derived by manual photointerpretation using false-colour stereo pairs and manual measurements of elevation differences. The orientation data of the images are determined by some ground control (measured by means of GNSS) and tie point measurements in the images. The orientation data of the images are the same both for the generation of the map and for the assessment.

C. Sample size determination and confidence intervals

Since per class user accuracy can be thought of as the probability p for a correct classification in a binomial experiment with x successes in n trials, one way to determine sample size is by specification of the desired width of the resulting 95% confidence interval (CI) for this proportion p. We will use likelihood ratio test (LRT) based confidence intervals for this proportion, because such intervals have better coverage probabilities than the standard Wald intervals often used in remote sensing [22]. Here, coverage probability of a CI refers to the proportion of times the CI will cover the true value of p (when thinking of an infinite number of hypothetical repetitions). The better coverage of LRT CIs is especially pronounced if p is near zero or one, respectively, or if the sample size is small. As an example, a 95% Wald CI in the situation where n=21 and p=0.85 will have an actual coverage probability of just 84%, which is far away from the nominal 95%, whereas a 95% LRT interval has a coverage probability of 94%.

LRT intervals for the proportion p are constructed by inversion of the LRT test for the hypothesis H_0: $p=p_0$ vs. H_1: $p \neq p_0$ [23]. That is, the lower and upper limit of a 95% LRT CI are found numerically as the boundary values of p_0 solving the inequality

$$-2 \cdot \ln\left(\Lambda_{n,x}\left(p_0\right)\right) \leq \chi^2_{0.95}. \qquad (4)$$

Here, $\Lambda_{n,x}(p_0)$ denotes the likelihood ratio between p_0 and the maximum likelihood estimate $\hat{p} = x/n$, i.e.,

$$\Lambda_{n,x}\left(p_0\right) = \frac{p_0^x \left(1-p_0\right)^{n-x}}{\hat{p}^x \left(1-\hat{p}\right)^{n-x}},$$

and $\chi^2_{0.95}(1) \approx 3.841$ denotes the 95% quantile of the χ^2 distribution with one degree of freedom. Implementation of such LRT intervals can be found in the "R" package "binom" [24]. Assuming a worst-case user accuracy of about 60% for each class and a desired half-width of no more than w=10%, one can sequentially increase the sample size n until the confidence interval for p has a width smaller than $2 \cdot w$ when using the observation $x=0.6 \cdot n$. The "R" package "sampleSizeBinom" [25], which implements this procedure, obtains a required sample size of n=91 per class. Hence, our stratified accuracy sample when using 4 classes will totally consist of $91 \cdot 4 = 364$ units. In order to compute the previously described LRT confidence intervals also for the survey weighted accuracy measures, we use the procedure suggested by Rao and Scott [26], which is implemented as part of the function `svyciprop` in the "survey" package.

D. Kappa analysis

A further measure of agreement between two raters, who both classify a number of items into each of k categories, is Cohen's κ [21], which can be viewed as chance adjusted agreement measure defined as follows:

$$\kappa = \frac{\sum_{i=1}^{k} p_{ii} - \sum_{i=1}^{k} p_{i+} p_{+i}}{1 - \sum_{i=1}^{k} p_{i+} p_{+i}} \qquad (5)$$

Landis and Koch [27] characterize κ-values between 0.41 and 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1 as almost perfect agreement between the classification and the reference data. An estimate for κ is found by replacing p_{ii}, p_{i+}, and p_{+i} in (5) with the respective survey adjusted estimates. If one, however, ignores the sampling design underlying the numbers in the error matrix, and just uses the raw survey-unadjusted estimates for the p's, the obtained κ estimate will be biased. For example, if a particular poorly classified class is over-sampled, the obtained κ estimate might be too low.

For a STSI design, formulae for computing κ from the resulting error matrix can be found in [28]. In [28], the formulae are an instance of a more general technique for calculating point estimates and variances for transformations of survey weighted estimates (see, e.g., [29]).

To reflect uncertainty in the estimation of κ it is also prudent to specify a corresponding $(1-\alpha)\cdot 100\%$ CI. Again, [30] presents equations for computing the variance of the estimate, which enables the computing of Wald CIs for the stratified case. Note that the package "survey" contains the function svykappa to directly compute the survey weighted κ estimate and associated Wald CIs based on the general equations in [29] – this even works for more complex survey designs. It has to be remarked that the use of κ to assess classifier performance in remote sensing is somewhat debated [30], because κ is directly linked to the overall accuracy, but its scale is harder to interpret. Still, we report κ estimates here as a supplement measure to the overall accuracy, since κ just as the overall and user's accuracy is particularly easy to calculate with "R" – even in the case of a complex design of the accuracy sample. In summary, the principle in the assessment of the thematic accuracy is depicted in Figure 2.

As accuracy measures the survey weighted overall accuracy and Cohens's κ, as well as the producer's and user's accuracy for each class are selected. Each measure is supported by a corresponding CI in order to quantify the uncertainty in its estimation. Altogether, one of our points is that a per-class stratified sampling approach for the assessment of thematic accuracy is easily handled with "R". Such an approach provides reliable estimates for the accuracy of each class.

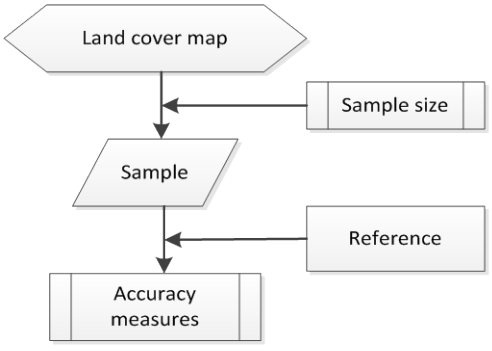


Figure 2. Principle of the applied assessment of the thematic accuracy.

VII. PRACTICAL EXAMPLE

A practical investigation is conducted for a residential area in Switzerland to evaluate the proposed approach. The area consisting of buildings, car ports, roads, parking lots as well as trees, hedges, and grassland is photographed from the air. The size of the test area is about 1.6 hectares.

The first land cover map comprises the four classes 'building', 'tree and hedge', 'grass', 'road and parking lot' only. The steps in the generation of the land cover map are depicted in Figure 3. True colour images are used to derive a very dense 3D point cloud by means of matching corresponding image parts. The point cloud is then transformed into a gridded DSM using interpolation techniques. Through a process of filtering, which removes elevations above ground as well as blunders, a DTM is generated. Then, the difference between the DSM and the DTM, the nDSM, can be derived. The DTM is also used for the production of a false colour orthoimage, which is further processed to a NDVI map containing two classes (vegetation, non-vegetation). NDVI and nDSM information is used to produce a point cloud with two attributes (nDSM, NDVI). The land cover map is produced by classification of

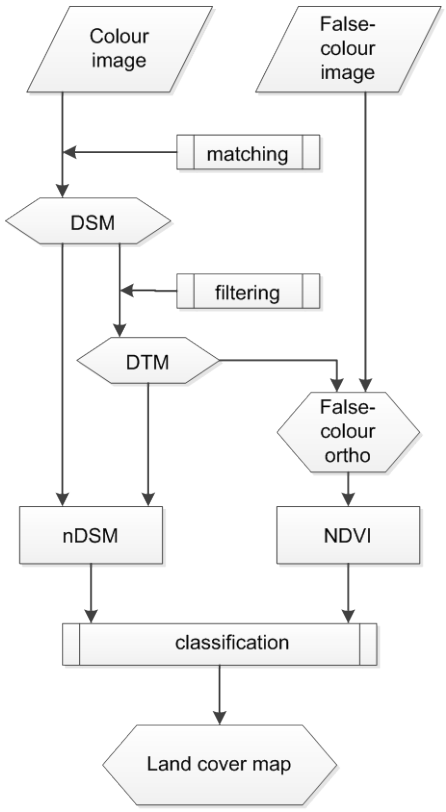


Figure 3. Steps in the production of a land cover map [1]. Colour and false-colour aerial images are used to derive accurate and dense height data (nDSM), which are supplemented with vegetation data (NDVI). Such an 'intelligent' 3D point cloud is input to a classification scheme, which generates the land cover map.

the point cloud with attributes. In this contribution, various methods of classification are investigated. The thematic accuracy of the land cover map is assessed by several accuracy measures including their uncertainty.

The classification program assigns a class to each cell. The cells (cubes) are plotted in 2D (3D) using different colours for the chosen classes. Some measures for quality assurance are undertaken during the generation of the land cover map in order to avoid errors and to obtain good results. This involves checking the georeferencing of all input data, their completeness, and the intermediate results. In the following sections, details of the data used, the tools applied, and the results of the assessment of thematic accuracy are given.

A. Used data

The RCD30 imagery includes four bands (red, green, blue, and near infrared) from which colour and false-colour images can be composed. Each image is about 6.4 Megabyte (Mb) in size. The images are taken with a GSD of about 5 cm. The radiometric resolution of each band is 8 bit or 256 intensity values. The images are geo-referenced and data of the geometric calibration (camera constant, position of principal point) are provided. False-colour images are used to derive reference data using stereovision. Furthermore, a map of building footprints is produced by means of digitizing a stereo-pair of colour images.

B. Software tools

For the generation of the 3D point cloud the Match-T program, version 5.4, of the Trimble/Inpho Company is used [31]. Filtering of the data is applied using filters of the professional program "DTMaster", version 5.4, of the same company. Manual editing may also be carried out by this program. The orthoimages in false colours are produced by Inpho's "OrthoMaster" program. The difference between the two elevation models (nDSM) is calculated by a simple C++ program written for the purpose (cf. Section IV.A). The generation of the land cover map is carried out in "R". Input is the nDSM and the NDVI map. The NDVI map is generated by the open source software LEOWorks [32]. The land cover map is visualized using "Quantum GIS" - an open source Geographic Information System [33]. The assessment of the land cover map uses the "DTMaster" program, which enables dynamic positioning to derived sample data.

C. Results

The calculation of the DSM (with a spacing of 0.25 m) yields a precision of σ_Z = 0.04 m. The absolute accuracy of the DSM is roughly checked by means of a few points, which had a maximum difference to the true values of 0.26 m. The binary NDVI map is generated by applying a threshold of T_{NDVI} = 0.1. The threshold for separating low and tall vegetation and non-vegetated objects is selected with T_{nDSM} =1.0 m. The result of the classification using these two inputs is depicted in Figure 4. The four classes are represented by different colours and are well separated from each other.

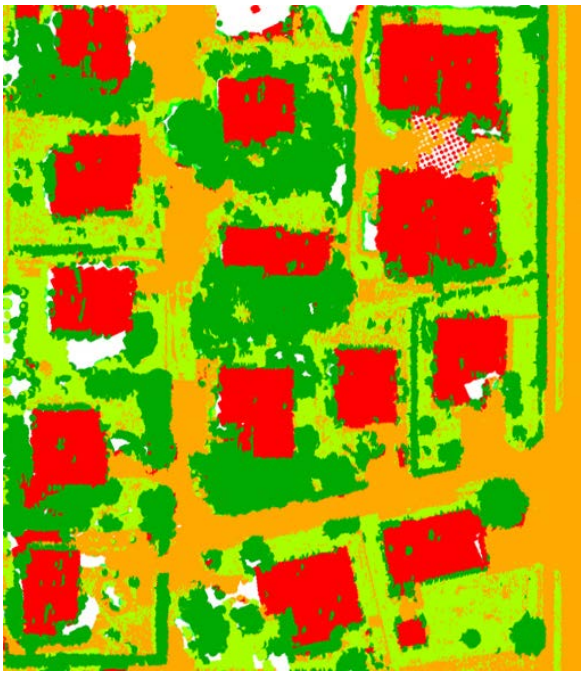


Figure 4. Result of classification in 2D. Legend: Red dots = 'building', brown dots = 'road and parking lot', bright green dots = 'grass', dark green dots = 'tree and hedge'.

Figure 5 shows an extract of the land cover map in 3D.

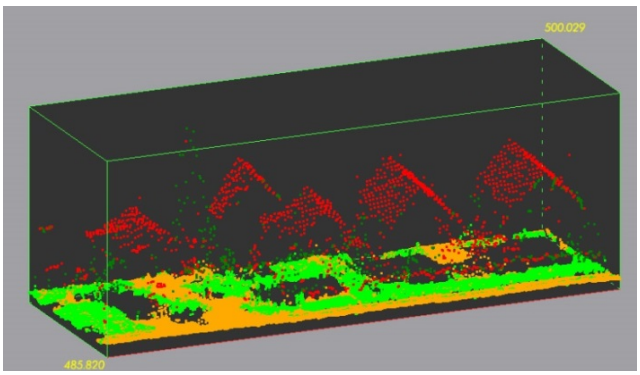


Figure 5. Visualization of spatial land cover map in 3D.

TABLE III. ERROR MATRIX OF THE DERIVED LAND COVER MAP WITH FOUR CLASSES FOR THE STRATIFIED SAMPLE.

class \ *reference*	*b*	*r*	*t*	*g*	*row total*
building	**72**	7	9	3	91
road & parking lot	3	**79**	1	8	91
tree & hedge	8	2	**64**	17	91
grass	0	2	8	**81**	91
column total	83	90	82	109	364

D. *Assessment of the thematic accuracy*

The overall number of classified cells in the test area when using four classes was 255222 (b), 254890 (r), 155819 (t) and 241408 (g), which gives a distribution of 28%, 28%, 17%, and 27% of the total area. 364 DSM-cells are selected based on the STSI design and are compared with the reference as shown in the error matrix of Table III. The resulting survey weighted overall accuracy of the classification is 82% (95% CI: 78-86%).

The user's accuracy of the four classes can be found in Table IV. The survey weighted κ estimate is 0.76 (95% CI: 0.71-0.81). When only two classes (i.e., 'vegetated' and 'non-vegetated') are considered, the overall accuracy is 91% (95% CI: 88-94%).

E. *Refinement of the classification*

The achieved results may be improved by selecting another strategy and definition of thresholds for NDVI and nDSM. Leaving the binary split of NDVI at 0.1, we refine the subdivision of nDSM by using splits at both 1m and 3m. Figure 6 shows the resulting six classes, i.e., the vegetated classes 'grass', 'hedge and bush', and 'tree' and the non-vegetated classes 'road and parking lot', 'wall and car port', and 'building'. According to the classifications, the percentage of the classes of the total area are 27%, 9%, 8%, and 28%, 4%, 24%, respectively.

TABLE IV. USER'S ACCURACY OF THE DERIVED LAND COVER MAP WITH FOUR CLASSES

class	*accuracy*	*95% CI*
building	79%	70%-87%
road & parking lot	87%	79%-93%
tree & hedge	70%	60%-79%
grass	89%	81%-94%

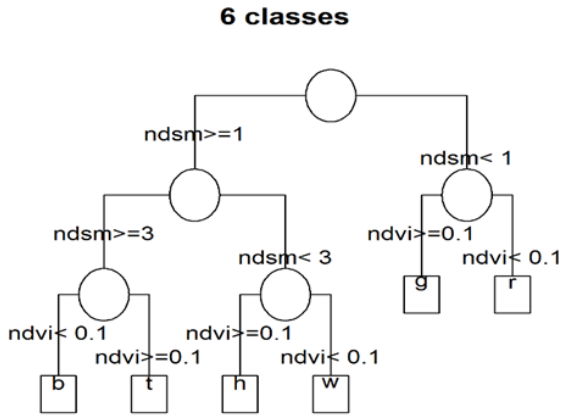


Figure 6. Classification scheme for a land cover map with six classes. The abbreviations are: b='building', t='tree', h='hedge and bush', w='wall and car port', g='grass'.

The derived land cover map with six classes is depicted in Figure 7; white areas indicate areas, which are not classified. The error matrix obtained from using stratified sampling is displayed in Table V. In this case, the overall accuracy is 86% (95% CI: 83-88%). User's accuracy for each class is shown in Table VI and the survey weighted κ estimate is 0.82 (95% CI: 0.78-0.85). When only two classes ('vegetated' and 'non-vegetated') are considered, the overall accuracy is 95% (95% CI: 93-97%).

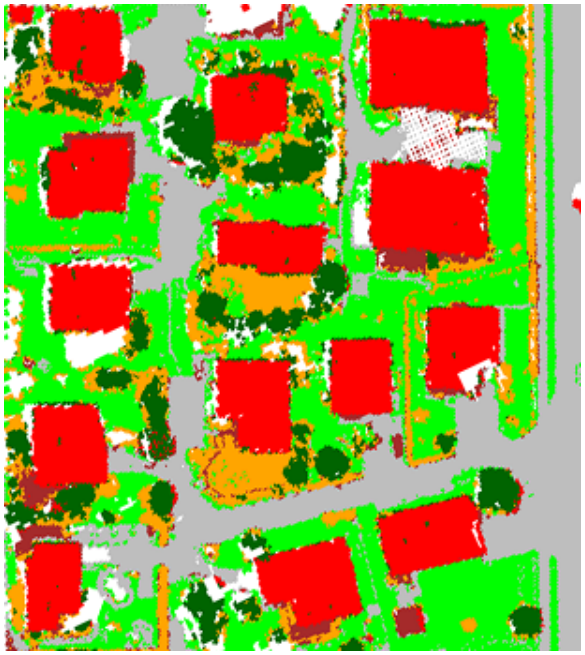


Figure 7. Land cover map with six classes. Legend: Red dots = 'building', grey dots = 'road and parking lot', bright green dots = 'grass', dark green dots = 'tree', dark yellow dots = 'hedge and bush', brown dots = 'wall and car ports'.

TABLE V. ERROR MATRIX OF THE DERIVED LAND COVER MAP WITH SIX CLASSES.

Class \ *Reference*	***b***	***h***	***g***	***r***	***t***	***w***	***row total***
building	**90**	0	0	0	1	0	91
hedge & bush	0	**41**	29	5	16	0	91
grass	1	11	**75**	4	0	0	91
road & parking lot	0	1	4	**86**	0	0	91
tree	3	2	8	1	**76**	1	91
wall & car port	5	8	1	13	2	**62**	91
column total	99	63	117	109	95	63	546

Another possible improvement is the use of more advanced classification methods than the current classification trees with manually derived rules for the tree splitting. We, therefore, investigate the use of classification and regression trees [34] with automatically derived splits based on recursive partitioning and as implemented in the "R" package "rpart" [19]. The classification of each cell is based on the continuous measurement values of both nDSM and NDVI.

In order to take the sampling design into account during the classification we weight each cell *i, i=1,…,n,* in the accuracy sample by its inverse inclusion probability, i.e., in our case of stratified simple random sampling the weights are $w_i=N_{c(i)}/n_{c(i)}$, where *c(i)* denotes the class of cell *i*. An overall accuracy of this classification procedure is then calculated by *n*-fold cross-validation, i.e., for each cell *i* in the accuracy sample a classification tree is fitted using the other *(n-1)* weighted cells as training set, then the obtained tree is used to classify cell *i*. Overall accuracy in its survey weighted form (as explained in Section VI.A) is then computed from these predictions. Such fitting of classification trees and accuracy measure computations can conveniently be performed using the "R" package "ipred" [35].

TABLE VI. USER'S ACCURACY OF THE DERIVED LAND COVER MAP WITH SIX CLASSES

class	***accuracy***	***95% CI***
building	99%	95%-100%
hedge & bush	45%	35%-56%
grass	82%	74%-89%
road & parking lot	95%	88%-98%
tree	84%	75%-90%
wall & car port	68%	58%-77%

In the case of four classes and the accuracy sample of *n*=364, the resulting overall accuracy of the automatically fitted regression trees is 77% (95% CI: 72-81%) as compared to 79% (95% CI: 74-83%) of the manual tree.
Note: The overall accuracy of the manual tree is slightly different compared to the numbers presented in Section VII.D, since the NDVI values are here calculated from an image with a higher resolution (3300 pixel x 3200 pixel instead of 802 pixel x 828 pixels of the NDVI-map derived by the program "LEOWorks").

In case of six classes with n=546, the overall accuracy is 81% (95% CI: 78-84%) as opposed the 82% (95% CI: 79-85%) of the manual tree. Altogether, the two results show that the gain in accuracy for our test area is not substantial when using the automatic classification trees trained from the accuracy sample. We, therefore, focus the presentation on the more easily interpretable manual tree.

VIII. DISCUSSION

The achieved overall survey weighted accuracy of the land cover map with six classes is 86%. Regarding the important user's accuracy, the result for class 'building' is correct with 99% and of class 'road and parking lot' with 95%. This is a very good result because simultaneously the survey adjusted producer's accuracy for the two classes is 97% and 92%. The results for classes 'grass' and 'tree' are also good with 82% and 84% (corresponding producer's accuracy of 82% and 76%). The derived survey weighted κ value is 0.82, which is better than the 0.76 at four classes. Using six classes improves the overall accuracy.

Automatically fitted classification trees are a computationally intensive supervised learning approach, which use more flexible splits and thresholds to perform the classification. However, these methods require a sufficiently large training set to work well. In our test example, the tree splitting rules are initially manually selected based on a-priori contextual information (e.g., a house has a certain height) and the resulting classification trees actually perform comparable to the more sophisticated machine learning approach, while being easier to interpret. One reason for this is that contextual knowledge is not captured by the purely data driven machine learning procedure - in case of small accuracy samples this makes a difference. A way to improve the automatic procedure will be to use a Bayesian framework for the classification trees where contextual information about splits is reflected as data augmented prior cases. However, when training sets are larger or contextual prior information less clear (e.g., in case of many diffuse classes or many indirect input variables), the automatic classification trees method and its refinements (e.g., classification tree ensembles) may provide superior classification results. However, our aim was not to fully explore the details of such complex classification for our test area, but rather to show

that normalized DSMs and NDVI maps provide important additional information for land cover classification and that training and subsequent accuracy assessments of the classification can be refined using stratified sampling without much extra software effort. Future work could focus on adding extra cell-based attributes and on different machine learning algorithms such as support vector machines [36].

The displacements of elevated objects in standard orthoimages could be avoided by means of using true orthoimages instead of standard orthoimages when compiling the NDVI map. The use of true orthoimages requires digital building models (DBMs), which are seldom available because their production is very expensive.

IX. EVALUATION

In this paper, elevation data of high density have been derived from aerial images. The applied high-resolution multispectral images enable the generation of a vegetation map and the generation of heights above ground. By means of a combined use of height and vegetation data a land cover map can be produced with a high degree of automation. Such a map is the graphical output of a point cloud with two attributes (height above ground and vegetation index).

Land cover maps with even more classes than the six investigated in our work can be generated when a finer partitioning of objects attributes is used or further attributes characterizing objects are added. For example, more intervals in the height above ground can differentiate various types of vegetation. In addition, water areas can automatically be extracted from the near-infrared band of the multispectral imagery.

An important prerequisite for good results of land cover maps is the quality of the applied imagery. Reference [1] investigates images used in this work. It is concluded that the images of the used camera have similar good results in the point spread function as large-format aerial cameras. The use of a medium-format digital camera (which is considerably less expensive than a large-format camera) is also a novel approach for the generation of land cover maps. Furthermore, its low weight and dimension enable the installation in helicopters and unmanned airborne vehicles. The use of medium-format aerial cameras instead of large-format cameras seems to be a general trend in the field of mapping and GIS.

DTMs, DSMs, nDSMs, or 3D point clouds may already exist for large areas. The generation of land cover maps may use such elevation data of existing databases. However, quality checks have to be carried out in order to determine whether the data are fit for purpose. High density, high positional accuracy, and completeness of the elevation data are prerequisites of the presented approach.

X. CONCLUSION AND FUTURE WORK

New advanced aerial cameras and new processing software make the automatic generation of urban land cover maps more efficient. The additional use of elevations yields better results in the classification. In this contribution, the assessment of the thematic accuracy of a test area is carried out by stereo observations of false-colour images for the derivation of reference values. The achieved good results at the manual tree classification (86% overall accuracy) as well as at the automated classification based on cross validation (81% overall accuracy) are possible, because accurate and dense elevation and vegetation data can be derived from the high-resolution multispectral imagery. The selected approach is, thereby, superior to classification from low-resolution satellite data. The open source software "R" provides all the tools needed for performing statistical prudent classification and accuracy evaluation of land cover maps. The updating of topographic data bases receives new possibilities to improve existing procedures.

In summary, we suggest to try out the proposed method on larger test areas – possibly also comprising of additional urban structures – in order to gain more insights about the scalability and robustness of the method.

ACKNOWLEDGEMENT

We thank Hexagon/Leica Geosystems for providing aerial image data and Trimble/Inpho for lending the software packages "Match-T" and "DTMaster" for deriving and editing point clouds. The comments of several anonymous reviewers helped to improve the manuscript.

REFERENCES

[1] J. Höhle, "Generation of land cover maps using high-resolution multispectral aerial cameras", Proc. GEOProcessing, 2013, pp. 133-138.

[2] J. Höhle, "DEM generation using a digital large format frame camera", Photogrammetric Engineering & Remote Sensing, 75:1, 2009, pp. 87-93.

[3] R. G. Congalton and K. Green, "Assessing the accuracy of remotely sensed data", CRC Press, 183p., 2009, ISBN 978-1-4200-5512-2.

[4] T. Hermosilla, L. A. Ruiz, and J. A. Recio, "Land-use mapping of Valencia City area from aerial images and LiDAR data", Proc. GEOProcessing, 2012, pp. 232-237.

[5] X. Meng, N., Currit, L. Wang, and X. Yang, "Detect residential buildings from lidar and aerial photographs through object-oriented land-use classification", Photogrammetric Engineering and Remote Sensing, vol. 78, no. 1, 2012, pp. 35-44.

[6] T. Novack, T. Esch, H. Kux, and U. Stilla, "Machine learning comparison between WorldView-2 and QuickBird-2-simulated imagery regarding object-based urban land cover classification", Remote Sensing, vol. 3, no. 10, 2011, pp. 2263-2282.

[7] N. Wolf, "Object features for pixel-based classification of urban areas comparing different machine learning algorithms", Photogrammetrie, Fernerkundung, Geoinformation, 2013, no 3, pp. 149-161.

[8] N. Champion, "Detection of unregistered buildings for updating 2D databases," EuroSDR Official Publication, no. 56, 2009, pp. 7-54.

[9] L. Elmansouri, "Object-based approach and tree-based ensemble classifications for mapping building changes", Proc. GEOProcessing, 2013, pp. 54-59.

[10] L. Halounova, K. Veprek, and M. Rehak, "Geographic information systems models of 40-year spatial development of towns in the Czech Republic", Proc. GEOProcessing, 2011, pp. 75-80.

[11] M. Gruber, M. Ponticelli, R. Ladstädter, and A. Wiechert, "UltraCam Eagle, details and insight", Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B1, 2012, pp.15-19.

[12] K. Jacobsen and K. Neumann, "Property of the large-format digital aerial camera DMC II", Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XXXIX-B1, 2012, pp. 21-25.

[13] R. Wagner, "The Leica RCD30 medium-format camera: imaging revolution", Proc. Photogrammetric Week '11, Wichmann Verlag, 2011, pp. 89-95.

[14] R Development Core Team, "R: A language and environment for statistical computing", R Foundation for Statistical Computing, Vienna, Austria, 2013, ISBN 3-900051-07-0, http://www.r-project.org/ (accessed 10.11.2013).

[15] N.Haala, H. Hastedt, K. Wolff, C. Ressl, and S. Baltrusch, "Digital photogrammetric camera evaluation – generation of digital elevation models", Photogrammetrie, Fernerkundung, Geoinformation, 2010, no.2, pp. 99-115.

[16] H. Hirschmüller, M. Buder, and I. Ernst, "Memory efficient semi-global matching", ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. I-3, 2012, pp. 371-376.

[17] T. Heuchel, "Trimble Match-T DSM", Proc. of the EuroSDR workshop on 'High Density Image Matching for DSM Computation', EuroSDR publication nr. 61, 2012, 39 p.

[18] T. Lumley, "Survey: analysis of complex survey samples". R package version 3.28-2,
http://cran.r-project.org/package=survey
(accessed 10.11.2013).

[19] T. Therneau, B. Atkinson, and B. Ripley, "rpart: Recursive partitioning", R package, version 4.1-1.
http://cran.r-project.org/package=rpart (accessed10.11.2013).

[20] E. Glaab, J.M. Garibaldi, and N. Krasnogor, "vrmlgen: An R package for 3D data visualization on the Web", Journal of statistic software, 2010, vol. 36, issue 8, pp. 1-18, http://www.jstatsoft.org/v36/i08/paper (accessed 10.11.2013).

[21] J. Cohen, "A coefficient of agreement for nominal scales", Educational and Psychological Measurement. vol. 20, no. 1, 1960, pp. 37-40.

[22] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion", Statistical Science 2001, vol. 16, no. 2, pp. 101–133.

[23] G. A. Young and R. L. Smith, "Essentials of statistical inference", Cambridge University Press, 2005.

[24] S. Dorai-Raj, "binom: Binomial confidence intervals for several parameterizations", R package version 1.0-5, 2009, http://cran.r-project.org/package=binom
(accessed 10.11.2013).

[25] M. Höhle, "binomSamSize: Confidence intervals and sample size determination for a binomial proportion under simple random sampling and pooled sampling", R package version 0.1-2, 2009.
http://cran.r-project.org/web/packages/binomSamSize/
(accessed 10.11.2013).

[26] J. N. K. Rao and A. J. Scott, "On chi-squared tests for multiway contingency tables with cell proportions estimated from survey data", The Annals of Statistics, vol. 12, no.1, 1984, pp. 46-60.

[27] J. Landis and G. Koch, "The measurement of observer agreement for categorical data", Biometrics, vol. 33, 1977, pp. 159-174.

[28] S. V. Stehman, "Estimating the kappa coefficient and its variance under stratified random sampling", Photogrammetric Engineering & Remote Sensing, vol. 62, no. 4, 1996, pp. 401-407.

[29] C-E. Särndal, B. Swensson, and J. Wretman, "Model assisted survey sampling", Springer, 1992.

[30] G. M. Foody, "Sample size determination for image classification accuracy assessment and comparison", International Journal of Remote Sensing, 2009, vol. 30, no. 20, pp. 5273-5291.

[31] T. Heuchel, A. Köstli, C. Lemaire, and D. Wild, "Towards a next level of quality DSM/DTM extraction with Match-T", Proc. Photogrammetric Week '11, Wichmann Verlag, 2011, pp. 197-202.

[32] ASRC, "LEOWorks, version 4.0," 2011. http://leoworks.asrc.ro (accessed 10.11.2013)

[33] QGIS, "User Guide of Quantum GIS program, version 1.8.0", 2013, 216 p.,
http://download.osgeo.org/qgis/doc/manual/qgis-1.8.0_user_guide_en.pdf (accessed 10.11.2013).

[34] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J Stone, "Classification and regression trees", Wadsworth, 1984.

[35] A. Peters and T. Hothorn, "ipred: Improved predictors", R package version 0.9-1, http://cran.r-project.org/package=ipred (accessed 10.11.2013).

[36] C. Hung, L. S. Davis, and J. R. G. Townshand, "An assessment of support vector machines for land cover classification", International Journal of Remote Sensing, 2002, vol. 23, no. 4, pp. 725–749.

Process Mining in a Manufacturing Company for Predictions and Planning

Milan Pospíšil
Department of Information Systems
BUT, Faculty of Information Technology
Brno, Czech Republic
ipospisil@fit.vutbr.cz

Vojtěch Mates
Department of Information Systems
BUT, Faculty of Information Technology
Brno, Czech Republic
imates@fit.vutbr.cz

Tomáš Hruška
Department of Information Systems
BUT, Faculty of Information Technology
IT4Innovations - Center of excellence
Brno, Czech Republic
hruska@fit.vutbr.cz

Vladimír Bartík
Department of Information Systems
BUT, Faculty of Information Technology
Brno, Czech Republic
bartik@fit.vutbr.cz

***Abstract*—Simulation can be used for analysis, prediction and optimization of business processes. Nevertheless, process models often differ from reality. Data mining techniques can be used to improve these models based on observations of a process and resource behavior from detailed event logs. More accurate process models can be used not only for analysis and optimization, but also for prediction and recommendation as well. This paper analyses process models in a manufacturing company and its historical performance data. Based on the observation, a simulation model can be created and used for analysis, prediction, planning and for dynamic optimization. Focus of this paper is in different data mining problems that cannot be solved easily by well-known approaches like Regression Tree.**

Keywords - business process simulation, business process intelligence, data mining, process mining, prediction, optimization, recommendation, association rules, genetic algorithms.

I. INTRODUCTION

Classic simulation can be used for the analysis of business processes. It is useful to test many variations of processes, measure the effects and then choose the optimal process settings. Thus, the process can be redesigned. It is possible to change resource allocation and search for the most optimal configuration with respect to context-based requirements (price, effectiveness, customer satisfaction, etc.). The current process configuration can be tested to discover how many cases it can handle over periods of time.

These models can be built manually but this is time consuming and error prone. The main disadvantage is that this approach cannot be used for predictions of operational decision, but only for strategic decisions if there exist some dependency on case attributes (see later). The operational decisions are important for internal logistics purposes. The casual models have some simplifications – for example overall probabilities of routing and naive execution time of the task. These parameters are set with respect to on long observation of processes, so they can work in a long-term simulation for strategic decisions. Nevertheless, operational decisions require short-term simulation. These two simulation approaches differ significantly. The short-term simulation starts in the current state of the process with allocated resources, cases in progress with known parameters and with waiting cases to handle. Routing probabilities and execution times can differ significantly for different case variables, therefore, mining of deeper dependencies is needed for better solution.

For example, let us assume a repair process. There are two tasks – repair of a basic item and repair of an advanced item, repair of a basic item is executed in 90% of cases and repair of an advanced item only in 10% of cases. Execution time for a basic item is about one hour and execution time for an advanced item is about eight hours. If our current case has attributes and these attributes lead to advanced repair with 80% probability, classic approach using overall routing probabilities is not precise enough to be used. And there is one another problem – execution time of a task is also influenced by case attributes – some case attributes may lead to longer execution time. Resources have to be also taken into account, e.g., some people work faster and some of them slower.

Predictions, recommendations, and dynamic optimizations could be accomplished by operational simulation. The system can warn us that some cases will be probably late, based on comparison with historic performance data. Then, some different scenarios can be simulated and evaluated. After that, the system can recommend us actions and provide dynamic optimization of current running cases – for example; assigning extra

resources from a non-critical case to critical or use a different sub-process – if we have a slower and cheaper or faster but more expensive variation.

This paper analyses processes of a manufacturing company. Simulation model is built using process mining and it is used for predictions. Based on these predictions, managers can change priorities (reallocation of resources) or better plan their storage space, because working front is known and, therefore, they can better predict manufacturing time. Building of simulation model from discovered model is beyond that paper, because analysis of data is also quite hard problem because of many real issues that need to be solved – that leads to some different problems. So this paper is mainly about analysis of data and then, these results could be used for the simulation and for other problems as well.

This work is an extension of our previous research in process mining and simulation field [1]. Paper is organized as follows. Section II describes related papers. Section III is overview of whole idea with some topics as utilization in real processes or problems with data preparation. Our real manufactory is described in Section IV. Then, three different approaches are described in Sections V, VI and VII. It is because different types of real problems that are needed to achieve our goal. Section V describes prediction using standard classification (like regression trees). Section VI solves the same problem but with variable feature vector length. This type of problem could be solved by classification methods, but many of them has problem with variable vectors. So we chose Association Rules to solve this. Both sections are based on existing methods with some extensions to better fit our problem. These extensions are described quite in general, not only for our company. Section VII solves another type of common problems – unmeasured processes. These are processes that are not fully measured and we propose some solutions to deal with it. Also, some further research is consulted. Section VIII is about analyzing errors. Previous sections are about predicting execution times, but errors are also important. Section IX is our summary of experience in real manufacture and about problem with measurement, because it is costly. Last section is conclusion. Experiments are provided in all of three main sections (V, VI, VII and partly VIII) for better intelligibility.

II. Related Work

Data mining techniques can be used in Business Process Management. The research area is referred to as Process Mining [2][3][4][5][6]. It is focused on analysis of information from event logs that were produced by business processes. Process discovery (Figure 1) is one of the methods and it is able to find a process model from an unknown process using many sequence examples of tasks and case parameters.

Except process model, also decision rules, social networks [3][7][8] and simulation models [7][8][9][10] could be discovered.

Resource behaviour is also a point of interest [11][12][13]. Example of simulation model [7] is depicted in Figure 2.

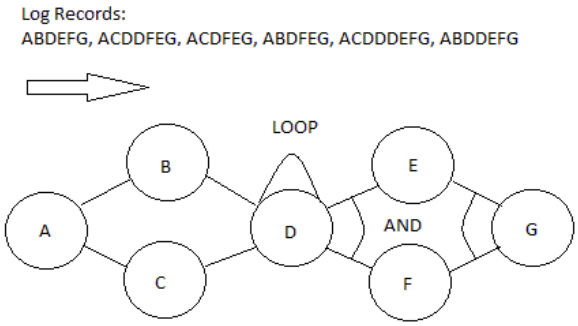


Figure 1. Process discovery. It is possible to discover a process model from the trace log.

It is possible to see routing probabilities and decision rules (decision rules are used when case attributes are known – that leads to better routing rules) and it is possible to see time distribution of tasks.

Some other research on process prediction was published in [8][14][15][16]. Wetzstein [15] used decision trees to analyse process performance (see Figure 3). As it can be seen, if the response time of a banking service is higher than 210, KPI (key performance indicator) is always violated. If customer id is 1234, manager can observe process bottlenecks and try to make banking service faster or find out why the customer 1234 has problems.

Grigori [16][17] uses similar approach used not for analysis but for predictions. Huge classifier is learned based on case attributes, start time and end time of task execution. Classifier can predict final execution time of a case based on case parameters and time information from executed tasks. Evaluation of that approach compared to our approach is discussed in [10]. In addition, our work uses similar approach as [15][16][17] but it combines it with process mining.

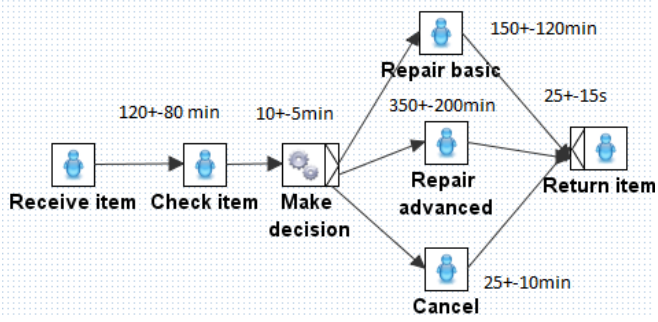


Figure 2. Simulation model. Classic simulation model is enhanced by decision rules. Decision rules can make our routing probabilities more precise, because they depend on case attributes.

Finally, when we discover deeper dependencies between routing rules and execution time of cases, we can use it for simulation related to decision support [10].

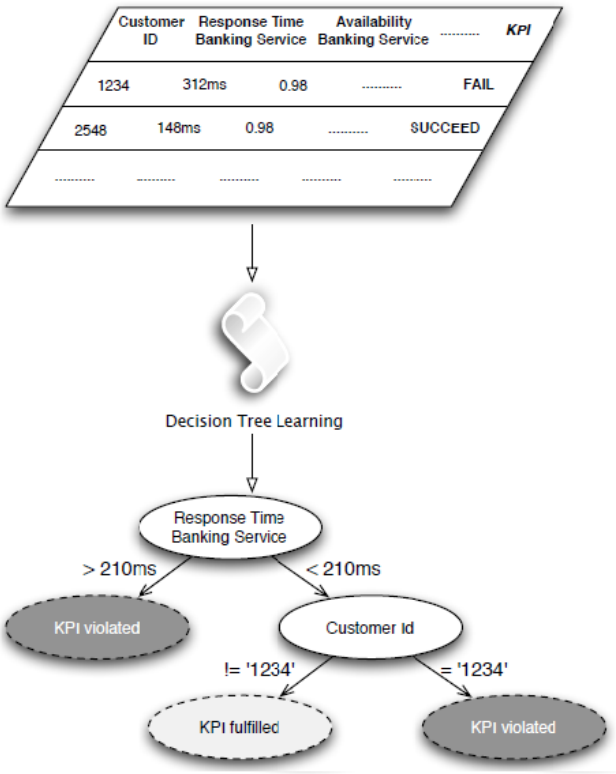


Figure 3. Process performance analysis (taken from [4]). Decision tree is used to discover factors that lead to KPI violation. We can see that KPI is violated when response time of a banking service is larger than 210.

Our previous work [1] is an extension of papers [7][8][10] and it adds some important features, some of them inspired by papers [15][16][17]. For example, execution time of cases could be also predicted by a classifier such as decision rules. This paper shows that our theory of [10] can be applied in a real large manufactory company. It also adds some additional theory in Section III.

III. More Precise Simulation Model

As it is said in [10], there is a demand on building more precise models than the one described by papers [7][8]. We will describe steps needed to accomplish it.

A. Process Discovery

If a process is not known, it is possible to discover it using process discovery techniques [4][6]. However, only process discovery is not sufficient to build simulation model. If an explicit model is not present, it is possible to discover it, but the precision of the model will be lower than the explicitly given by a real model. In some companies, discovered model could be more precise than an official model but it is because these companies do not have their models well-structured in many cases. This is not the case for manufacturing companies where prediction and usage of short time simulation is considered to be better.

B. Decision Mining

Decision mining is based on discovering routing rules in OR split nodes. These rules could be also available but sometimes they are not applicable. Let us assume that a routing rule is based on one parameter, which is inserted into the system just before the decision. Thus, our predictor will know the next path only in the time of decision – this is a useless prediction. In these situations, decision mining has to be used. The topic of decision mining is described in [7], [8][10]. Classifier is learned on training data where inputs are case attributes and output is the next path in process. Our work [10] describes another problem, which is missing attributes or 100% precise attribute known in the time of decision inserted by human (described earlier). If some attributes are missing, classic classifiers will not work in a proper way. If there is a 100% precise attribute then classifier is based only on that attribute. Solution is the same for both problems – it is necessary to build several classifiers for several milestones of the process – from the start (only subset of case attributes are known) to the end (all attributes are known).

C. Execution Time of Tasks

Execution time of tasks is the most important issue in the short-term simulation. Process model and routing rules are important as well. However, in companies with predictable business processes (especially manufacturing companies), control flow and routing rules are used to be formalized and nearly 100% precise.

Execution time of tasks will be described precisely in Sections V, VI and VII where different approaches for different problems will be introduced.

D. Usage of analysis and simulation

There are several ways to use our methods in practice. We must realize that we must do two tasks. First task is the analysis of historic data, which is the main focus of the paper. Second task is to build the simulation model for predictions based on previous data discovery. Second task is not the main topic of this paper, but it is the main goal. Now, we will describe utilization of both steps.

1) Prediction, Planning, Reccomendation

First usage is obvious and we described it in Section I. If we know the task execution times (does not matter if it is the result of data mining or manual measurement), we can better plan our whole work, material flow, inventory, machine and resource utilization. Using simulation, we can evaluate multiple plans and choose the most suitable of them. Also, we can monitor running process and check if some problem is going to come in future. Of course, with some probability – there must be multiple simulation runs.

There is other utilization in scenario testing. Managers can make some change in the process (better machine) to test what is the influence and cost of the change.

2) Analysis

Because the first step is about data mining, we can use these discovered information for another use. If we know what case attribute causes what time and variation, manager could ask for most influential attribute combinations (low or

high time and variation). Variation is also very important because high variation suggest some production problems, which is not good for planning. With these tools, managers can focus on most problematic attribute combinations and try to solve them.

Another usage is for worker performance [13]. There is a problem with computation of worker performance, because it is dependent on attributes. Paper [13] describes it in further detail.

Error analysis is also an important part. Some attributes have higher error probability. Errors are important not only to predict execution time of tasks but also to predict routing between tasks – faulty product could be sent to repair and could not almost affect time of task where it occurred. Analysis of error is important for execution time of prediction and there is whole short section at the end of the paper that describes it more deeply.

Last usage is about analysis of changes. Suppose management of company bought new machine (or new working method). Was it really an improvement and how much? If we analyze both old and new task data, we can discover differences between them. For example, new machine could be overall better, but only for some attribute combination.

Analysis of time and variations based on attributes could be also good for advertising. Management could focus on products that have good attribute combinations for manufactory – in our door company, some doors are produced fast, some more slowly. If there is a big commercial on some more problematic doors, production should be late and it could harm name of the company.

3) Simulation based on historic data

This type of simulation is not for prediction, but for analysis. It is not the simulation as we know it, but only replaying of historic log data. Using this, we could analyse effect of changes on queues and much more interesting features. But this type of analysis is beyond the scope of our paper.

E. Problems in preprocessing step

Every data mining needs data cleaning and preprocessing. This is the step most dependent on data type. We should describe some problems we encounter in our practice.

1) Analysing quality of data set

First step we must always do is to gain information about data set reliability. We must analyse records with the same attributes, if they are available – they may not, but almost every data set contains some set of records with identical attributes. Then, we must compute variance of those similar records. For example, if there are twenty records with attributes A, B, C, we have to compute their variance execution times. If A, B, C = 250s, A, B, C = 300s, A, B, C = 350s, etc., then we have to compute variance from these numbers. If the result variance of nearly all of types of record set is not lower than the variance of whole data set (variance of all record times), the data set is useless. It means there is no dependency on attributes or some attributes are missing or there was some problem in measurement – last problem is the most probable in our experience.

2) Low and High Values

Beware extreme values in data sets, it means different set of scenarios. Extremely low values are probably errors that were discovered and sent to repair. Extremely high values are errors or poorly measured values. We must be careful, because sometimes break can affect time dramatically – suppose start of task was measured correctly, but then break occurred and then work continued – total task time will not be useful at all. So high values do not mean automatically error, if we want to analyze errors (see later), we need information about errors explicitly given in data. Another possibility is schedule of breaks.

3) "Half Measurement"

Another problem, sometimes solvable, is the "half measurement". Sometimes, there is only information about start or end of the process. Sure, we can deduct these times to get real time. Deduct means that if product A enters into a task and (for example) start time is measured, then product B enters the machine and another start time is measured. Then, if a task could contain only one product, we can compute Time as Start(B) – Start(A). It is easy, but it may not work. If we are deducting times (start or end), there must be quite high utilization of task (low waiting for product processing – pause). Every waiting could be considered as production because we have no idea if there was waiting or our product took more time to produce.

Fortunately, this problem could be sometimes solved, but only in tasks with low variation based on attributes. Global variation could be high, but there is a need that cases with identical attributes must be produced with little variation. If this is true, we could analyze different sets of records divided by attributes (every set of record has its own attributes and every records in set have the same attributes). The sets will have some execution times and time of every set will be ordered ascending. We will get something like this for some set of records with the same attribute. For example, there are twenty records for attributes A, B, C: 100s, 101s, 102s, etc., 110s, 112s, etc. 140, etc..If we suppose low variations of executions for the same attributes, first records are the real execution time whereas others are probably affected by waiting time.

Half measurement is not only burden that negatively affects accuracy of prediction but it can be also advantage. Measurement devices are also quite expensive so every saving counts.

IV. MANUFACTURING COMPANY

The manufacturing company is specialized in door production. Uniqueness of door is characterized by their attributes (about twenty) and based on these attributes, different operations are executed. Doors have different material, size, weight, different corner and edge types, different handle and glasses, etc. Every door has its ID and it can be modeled as a case. Doors are mainly manufactured in machines (tasks). Some machines work in parallel; some machines are bound to several tasks, thus these machines

must be treated as resources, because machine could be down or working. People are working with machines or in manual workplaces. Routing probabilities are 100% accurate, because doors with specific attributes must be manufactured only by a specific machine and with specific settings.

Resources are quite predictable, because they work on shifts and they are always available and planned several days ahead. The only unknown parameter is execution time of tasks that depends on case attributes – every case is modeled as one door, so case attributes are door parameters. Door parameters are known at the beginning of the process and are constant, so there is no need to build several classifiers for several periods of case execution [10]. Execution time also depends on people work rate, work queue and error rate (especially in manual workplaces), but this is issue beyond the paper.

Context-based predicting the execution time of tasks can help with several issues quite precisely. First, it is possible to decrease storage spaces, because they could plan execution order of cases in order to decrease waiting times. Our prediction decreases variances of execution time and thus logistics can have methods to plan storage spaces with better results when there is a low variance. They will also know if some doors will be probably late and for example they can respond to that changing priorities, resource allocation, etc. Another important issue is the analysis. Managers could measure which door types take long time to be produced and based on that, they can calculate their price more precisely. For logistics, execution time is not as much important as influence of variance of execution time. It is possible to measure which door types (based on parameters) have high variance. Process engineers can focus on those door types and try to find out the cause of high variance, or produce them only in situations (if it is possible to wait) when variance is not such important issue.

A. Usage in Company

Because our manufacturing company has quite a lot dependence on door attributes, it is important to discover them to better plan production. Order of different doors for production is also important, because some machines could produce some door types faster, some more slowly and we need to better balance the product flow.

V. Prediction of Tasks Execution Time Using Classification

The time deviation is sometimes high, but it can be decreased by data mining techniques. Thus, it is useful to examine data and find relationships between case parameters and execution time for each task in the process. This can be solved as a classification problem, where case parameters are considered as input attributes and execution time is considered as the target attribute.

A. Classification and Prediction Models

There are many kinds of classification models; every model has its advantages and disadvantages based on properties of data used for classification. Our problem is rather prediction than classification, but both issues are related and many models support both of them. One common definition is that prediction predicts future and classification works as pattern recognition. Other definition is that prediction works with numerical target attributes and classification with categorical target attributes. Prediction can be transformed to classification by transforming target attribute from numerical to categorical, where categories are intervals that covers whole domain.

In our case, we have 18 case attributes and one numerical target attribute. All case attributes are categorical. Even through there are also some of them numerical (width, height), but they are standardized to only few distinct values, so they can be considered as numerical or categorical depending on requirements of classification or prediction model. It is more difficult for prediction (even in our case) that target attribute varies even for cases with the same values of input attributes. This is typical for execution time, because work is performed not only by machines, but also by people and people do not often work in coherent speed.

High variability of door types is another problem. In our manufacturing company, it is possible to make millions kinds of doors, which causes problem in prediction, because it is difficult to obtain enough examples for prediction. Attributes can also contain high number of distinct values which correspond with high variability of door type (this is a problem for neural network classification).

In the next section, some prediction models will be described and its applicability is discussed.

1) Neural Network

We have tested Neural Network approach, but results have not been satisfactory. Neural Network was not able to learn. It was caused by the high number of input neurons - 303. Every categorical column had to be transformed into new columns. Every distinct value of that column created a new column, which holds 1 or 0 value. For example, the column corner has four distinct values – left, right, top, and bottom. It creates four new columns that can acquire value 1 only once for a row (for the columns that belong to one categorical column). That transformation was necessary, because neural network can handle only numerical attributes. Target attribute was divided into several intervals and every interval was modeled as a single output neuron.

We think that network was not able to learn because of high number of inputs compared to number of training examples and mainly because of the output variability of (even identical training examples had little different outputs). Thus, we think network is not sufficient for our problem because of high number of categorical attributes and variability of the target attribute.

2) K-Nearest Neighbour (KNN)

The method is based on a simple idea of finding several examples from training set closest to an input pattern. We simply computed number of differences between training example and input pattern. These differences (0 or 1, equals or not equals) were weighted. Weight of each attribute was computed by the same method described below by the regression tree. Higher weight means that attribute has higher influence on execution time and it is considered more

important. Then twenty nearest examples were given and mean, minimum, maximum and deviation of time was computed (we measured only mean, but deviation is also important in simulation and it is a good indicator of supposed reliability of prediction).

Results (Figure 4) were quite satisfactory (there is only subset of real workplaces). We have compared prediction to a simple algorithm – prediction based on mean of all execution times. The simplest predictor is the predictor that assumes mean value for every example. Differences in table are mean of all differences between real values and predicted values for every tested example. We have run the test with 600 examples and we have compared them to a dataset that contained approx. 10-20 thousands records for every workplace. Rating was computed as a ratio between difference computed by the algorithm and difference computed by mean. Thus, the result 0.5 means that we have decreased the variance of execution time of task by about 50%.

Figure 4 shows that some results are satisfactory, others are worse. For example, ratio of workplace A seems to be good, Workplace C is not as good as Workspace A. Nevertheless, it is not the problem related to the method, execution time does not rely only on attributes. It is for example the case of workplaces C, which perform packaging and that type of work is naturally quite independent of door types. .Workplace A is a machine that does not depend on resource skills, workplace B is a workplace with dependence on resource skills and workplace C is a manual workplace (packaging) that does not depend so much on door type, but on resource performance

3) Regression Tree

Decision tree is a popular model. It is simple, readable by human and quite fast. Precision has not been as satisfactory as results given by the K-Nearest Neighbor classifier. However, the classification speed is several hundred times faster. Regression tree is a decision tree with numerical target value. Nodes contain information about mean, minimum, maximum and deviation of predicted value. Learning algorithm is similar to decision tree, but selection of split nodes differs. We have numerical target attribute, therefore, algorithm can be as follows:

Input: A table, which contains a numerical target attribute.
Output: Decision powers of all attributes.

1. **For** each column *C*
2. **For** each distinct value v_i of column *C*
3. Take all *n* target values t_i of column grouped by current distinct value and compute their deviation σ_i.
4. Count the decision power of the column as:
   $DP(C) = 1 / (\Sigma\sigma_i / n)$, i.e., mean of all deviations.

Algorithm 1: Regression Trees

This algorithm is similar to entropy computation, which is computed for categorical target values. The deviation is closed to entropy because lower deviation points to better decision power. Computing of deviation can be also weighted by count of rows related to groups divided by distinct values of column – distinct value with more rows should be more important. We have tried both approaches, but no significant precision difference was observed, even maybe precision has been a bit lower. Algorithm described above works similar to ID3 algorithm. C4.5 algorithm has been also tried, however, no significant difference has been found. Post-pruning was based on removing nodes with low row count (every node corresponds to a subset of the whole data set), because nodes with low row count are not representative.

Regression Tree has had worse precision compared to K-Nearest-Neighbor (about 1.2 – 1.3 times worse), but it has had also several advantages. It is more readable to human and it can be used to examine some properties of tasks – for example, which combination of attributes affects execution time positively or negatively or which combinations of attributes have little ratio of prediction – that is represented by deviation of target values corresponding to some node of tree.

4) Regression Tree Forest

Regression Tree Forest is based on several Regression Trees. Obvious example can be the Random Forest. The Random Forest creates many decision trees (more than one hundred) using classic (ID3 or C.45) algorithm with several differences:

1. Every tree randomly selects subset of rows from a training set (about 2/3).
2. Every tree randomly selects subset of attribute columns (about 2/3).
3. Every tree is not pruned and full-grown.
4. Predictions are made by voting of all trees by computing mean.

It is known that Random Forest is a very precise model and it is still quite fast, because it is semantically similar to K-Nearest-Neighbor algorithm. However, learning time is quite long (it requires more than one hundred trees), we have found it not suitable for real-time decision support. However, we have tried some trade-of between Random Forest and normal Decision Tree. We created several (about ten) trees and enforced different first splitting column for every tree. Enforced columns were ordered by their decision power. Thus, first tree root node begins with the first (best) column; second tree root node begins with second column, etc. In addition, every tree randomly selects 70% of dataset and 70% decision attributes as it is used in Random Forest algorithm. Trees were pruned (opposite to Random Forest, which is not pruned) to about 10 rows in a node.

It should be stressed out that in normal Random Forest, result is computed by mean of all tree results. We have selected the best tree result by looking to the deviation of tree node. Best prediction could be measured by deviation of particular rows covered by a tree node. Node with lowest deviation wins. This rule was necessary because mean of all tree votes gave very unsatisfactory results – mainly because we have had only low count of trees compared to Random Forest.

Then, we designed another improvement – tree result (mean and deviation) was not computed only by looking at

the leaf node but also the parent node is taken into account. We have computed mean by both nodes mean values weighted by their deviation (if a child node had much better result – low deviation – than parent, its result will have bigger vote). That improvement has also been tested in a single ordinary Regression Tree and it increased precision too, but only slightly.

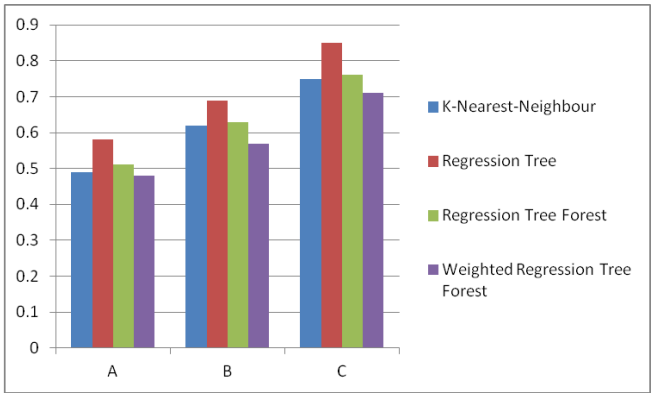


Figure 4. Experiments. Four methods were used on three workplaces. Note that a higher column means lower precision.

Our Tree Forest significantly improved accuracy of classifier, the ratio was only about 1.05 times worse compared to K-Nearest-Neighbor, however, it was the order of magnitude faster than K-Nearest-Neighbor, usability of which could be problematic in real time monitoring for all tasks due to performance issues. Similar results can be explained, because random forest works similar to K-Nearest-Neighbor. It returns items that are close (by attributes) to a predicted item, but it uses tree searching instead of searching in the whole table.

B. Execution Time and Resources

There is a little problem with resources. The resource information can be treated as a normal case attribute, because it has impact on execution time of task, but there can be some difficulties. For example, if we allow decision tree to build tree using resource attribute, final leaf will contain only records that were executed only by that resource. This could cause problems because sometimes it is better to look for more examples, even from another resource. However, if we do not have such training examples and resource performance does not differ too much from other resources, it is useful to look also to another resource records and consider them.

The second problem is related to dynamic changes. Even if the process is the same (e.g., technological process), workers performance changes over time. More experienced workers may be faster, thus our algorithm should be prepared for dealing with that kind of issues. We recommend the following method, which slightly improved prediction in our manufacturing company.

Suppose K-Nearest-Neighbour or Regression Tree (or Forest) classifier. All that classifiers could be implemented to return set of records rather than final prediction (mean and deviation). The result (mean, deviation) could be implemented over those records, but with different weights. First, records that belong to the resource, performance time of which is now predicting, should have bigger weight (for example two times higher) than other records. Second, these records (of our resource) should be considered in the time plan. Newest records should have also higher weights (for example two times higher than the oldest). Why it is not possible to take into account time plan also for other records (other resources)? Because it is very difficult to know about them enough information in order to take into account their improvement and skills compared to our resource. This could be issue to another paper.

C. Computing Variance

As we mentioned, low variance is important for good process of planning and material flow. It is useful to know variance of a task based on its attributes. Our tested method returns the set of examples and then computes means from them. It is not difficult to compute variance as well. But there can be a problem with different methods that do not return a set of examples, but only the final decision (not a regression tree problem, because leaves contains mean and also variance information even if algorithm is not built to return set of examples). For example, it can be problem with Neural Network, which is not built to return examples at all.

D. Test on Validation Data

Previous results were tested on training dataset. We have done another test using test dataset (about 20% of whole data) to prevent overfitting. Results were very similar to previous tests; therefore, we did not include them here. It is because methods like KNN and Regression Tree are quite robust to overfitting. The second reason is related to data – there is some variance even for records with the same attributes. So overfitting is not an issue here.

E. Summary

We have tested three tasks: One machine with little human interaction, second machine with manual work and third packaging with little dependence on door type, but with dependence on resource. As it was presented, Regression tree is always less precise than other methods, while Regression Tree Forest is as precise as K-Nearest-Neighbour, because it is like the optimized K-Nearest-Neighbour. Last method was the weighted Regression Tree Forest. As it has been shown, weighting on workplace A did not improve result at all, because machine works independent on resources and time (it does not learn to work faster). In Workplaces B and C, there was some improvement. Workplace C had the worst results, because packaging is not dependent on door type too much, but it is dependent on resource – we can see that weighting slightly improved performance.

VI. Prediction of Execution Time of Tasks using Association Rules

In some cases, there is a need to process non-relational data because sometimes the sizes of various cases can be different. Their main difference from classic relational data

is the fact that various records can contain various counts of values (case parameters). To predict execution time of a process, it is suitable to use the association rule based classification because data are similar to a transactional database. For our task, we will consider each case as a transaction.

A. Mining Association Rules

Association rules were first introduced by Agrawal et al. [18]. Mining association rules was primarily designed for usage in transactional data. Therefore, it is not any problem with discovering association rules from this kind of data. A lot of algorithms for mining association rules in transactional data have been developed. The Apriori algorithm [19] is probably the most famous of them because of its simplicity. On the other hand, the FP-Growth algorithm [20] proved to be much more efficient than the Apriori algorithm.

Association rules are most frequently used in market domain, typically for market basket analysis, where transactional databases are used. Here, the goal of mining association rules is to find rules of a form $A \Rightarrow B$ where A and B are sets of items. If this association rule is found, it is usually interpreted as: "If a transaction contains a set of items A, it is likely to contain a set of items B".

A formal description of the association rule mining problem is specified as follows. Let I = {i1, i2, …, in} be a set of items, which can be contained in a transaction. Let T = {t1, t2, …, tm} be a set of all transactions, where each transaction ti = {ii1, ii2, …, iik} is a set of items, where each item iij $\in$ I (for i $\in\langle 1,m\rangle$ and j$\in\langle 1,k\rangle$). If A is a set of items, a transaction t contains A in case that $A \subseteq t$. Then, an association rule is defined as an implication of the form $A \Rightarrow B$, where A, B $\subset$ I are sets of items, which are called itemsets. These sets must be disjoint. An itemset that contains k items will be called k-itemset.

Potential usefulness of a rule is usually expressed by means of two measures – support and confidence. The rule $A \Rightarrow B$ has a support s, if s% of transactions contains both itemsets A and B. It represents the probability of occurrence of the rule in the database. This probability can be expressed as

$$support\ (A \Rightarrow B) = P\ (A \cup B) \quad (1)$$

Confidence of the rule represents the strength of implication in the rule. It is the conditional probability that a transaction contains the set B provided that it contains the set A. This is expressed as

$$confidence(A \Rightarrow B) = P(B|A) = P(A \cup B)/P(A) \quad (2)$$

For each association rules mining task, a value of minimal required support and confidence must be specified. If the condition of minimal support and confidence is satisfied, the association rule is called a strong association rule. A frequent set is an itemset satisfying user-defined minimum support and strong rules are generated from frequent itemsets that satisfy also user-defined minimum confidence.

B. Association Rule Based Classification

The model described above can be also adapted for classification. The association rule based method was originally designed for classification of text documents into a set of predefined categories. Each text document is represented as a transaction – a set of words (terms), which occur together in the corresponding text document. Generally, a transaction is defined as a set of items. Therefore, the only required information is the occurrence of the term in a document.

Association rule based classification method was first introduced in [21]. The main advantage of it is that it provides a human understandable classification model in a form of association rules and good accuracy of classification. The Apriori-based algorithm is used to generate association rules.

The next method called CMAR [22] was based on a well-known FP-growth method for mining association rules, which is significantly faster than Apriori-based algorithms.

The method described in this paper is a modification of association rule based classification method designed for text classification. The original method described in [23] works with text documents represented as transactions (set of terms, which occur in the document). In the training phase, association rules are discovered from these transactions for each class separately by use of the Apriori algorithm.

For this classification method, we are focused only on a special form of association rules, which is usable for classification. The required form of a rule, which is a result of the training phase, is the following:

$$term_1 \wedge term_2 \wedge \ldots \wedge term_n \Rightarrow Cat, \quad (3)$$

where the antecedent of a rule contains a set of terms, which frequently occur together in documents that belong to category (class) Cat, which is contained in the consequent of an association rule.

We can see that the task of training phase of the classification method is to obtain a set of association rules for each category. This set of rules is forming the classifier.

While the set of categories is predefined and there is a set of documents belonging to each category, we can obtain a set of association rules separately for each category. For each category, a set of frequent itemsets is found with use of Apriori algorithm in a set of documents belonging to the corresponding category. Each frequent itemset is then associated with that category.

That is why the method is called ARC-BC (Association Rule Classification – By Category). This property allows to perform so-called multiple-class classification, because there can exist rules with the same antecedent and different category in the consequent. If it is necessary to assign only one category for each document, we have to decide according to the value of support or confidence of a rule. The association rules with lower value of support/confidence are omitted from the classifier.

If the set of rules is generated, it very often contains very high number of similar association rules. To reduce their quantity, pruning techniques can be used. The task is to find association rules that are more general and have higher confidence.

If we have a suitable number of rules for each category, we can use these rules to classify new objects (records, documents, etc.). We have to obtain a set of terms representing the new object. If there is an association rule that contains the same set of terms, the corresponding category is assigned to the object.

Usually, more than one association rule (for more than one category) is found for a classified document. It is necessary to set a dominance factor, which is counted as a sum of association rules' confidences. This allows getting a most dominant category or k most dominant categories for a document.

In [24], this method was adapted to use it for classification of classical relational based data but the experiments presented showed that it is not very accurate if the table contains more quantitative attributes, which must be discretized before association rules are going to be discovered. In our process mining task, discretization of input data is not required.

C. Usage in Process Mining

The main difference between our process mining task and the process described above is the target attribute, because we have to predict a numeric value of processing time in our data mining task. The task must be transformed into a classification task. The only possibility to do it is to discretize the target attribute. Regarding the discretization, we have to make two decisions – we have to choose the discretization method and the size of intervals created from the values of the target attribute. This choice depends primarily on the end user's requirements.

Sometimes there may be a need to make some post processing steps after the training phase of the method because the same frequent itemset can be obtained for different categories by this classification method. There are two possible post processing tasks resulting from this fact:

If the same frequent itemset is obtained for two adjacent categories (intervals in the antecedent of a rule), these two (or more) categories can be joined to form one association rule.

If this appears for two or more intervals that are not adjacent, we can omit the rule with lower value of support, if it is significantly lower. If the difference between two support values is not very high, we should keep all those association rules in the result of the training phase.

Discussion about this solution and presentation of some other interesting properties of this method will be contained in the experimental part of this paper.

D. Experiments

We have made some experiments with a real dataset from a door producer. The task was to predict time needed to make a door from the set of input attributes, such as material, size, weight, etc. The dataset consists of 17 input attributes and its size is 10000 records.

It was necessary to discretize the time attribute to apply the association rule based classification - the attribute was discretized into three categories (intervals), which represent low, standard and high time needed to produce the door.

1) Classification Accuracy

Our dataset is a relational table and, therefore, it is possible to compare association rule based classification with other classification methods.

The value of classification accuracy is between 75% and 80%, depending on values of minimum support and confidence thresholds given by the user. This value of accuracy is comparable with other classification methods such as naïve Bayesian classifier, decision trees or support vector machines. This leads us to a conclusion that our method is suitable to predict time also in data with variable number of attributes.

2) Examples of Association Rules

As the association rule based classifier provides a user understandable model, it is also possible to analyze the association rules for individual categories. Probably the most interesting for the user will be the category, which represents high produce time.

We can mention an example of association rule obtained by our method: door_construction = type_1 $\wedge$ edge_D=CH001 $\Rightarrow$ high_time.

This kind of association rule helps producer to plan his production and to find the reason of delays during the production process. We have obtained about 30 association rules for the category representing "high production time". Similar count of association rules has been obtained also for other categories.

It is also interesting to analyze association rules obtained for more classification categories. This denotes for higher dispersion of time value. This fact also means that attributes contained in those rules have less influence on the time of production. If these association rules are joint into one rule, we have association rules of the form from the following example: hardness = 1 $\wedge$ filling_type = DTA $\Rightarrow$ high_time $\wedge$ avg_time $\wedge$ low_time.

At the end, we can take only interesting associations (only those with class high_time or low_time) and then select all records that contain these attributes and compute mean and variance for them. For example: type_1 $\wedge$ edge_D = CA003 $\Rightarrow$ 230s $\pm$ 100s.

VII. UNMEASURED PROCESSES

Unmeasured processes are processes with known process model, however, only the length of execution of the whole process is known. This case can be related to many companies, because measurement of every process step is expensive. But even if we do not know the exact time for every task, we are able to discover some of them if enough data is available.

Unmeasured processes could be static or dynamic. Static processes are processes that have always the same tasks in their process model. For dynamic processes, the process

model is built during the runtime based on attributes of a process instance (case).

Analyzing hidden processes could be easily topic for whole new paper, thus we will describe only situations related to our manufactory company. But our experience should be applicable for many other companies with the similar problem.

We give theory only for dynamic processes because there is one problem with static ones – if every process instance is the same, we cannot compute what is inside the process, because it is the changes what allow us to find something valuable. On the other hand, static processes are more predictable and they could be treated as atomic task.

A. Sequential Dynamic Processes

Sequential dynamic processes are processes with sequential flow but every process instance contains different tasks. One process instance could contain tasks A, B, F, H, another one could contain C, E, I. We know duration of the whole process instance and names of executed tasks (and their order, if it is important). So our information can look as follows:

A, B, F, H	=	213s,
C, E, I	=	170s,
etc.		

Our objective is to assign average time duration to every task in order to best fit the execution time of the whole process. It is obvious that total execution time is the sum of lengths of all tasks in the case. If A = 100s, B = 60s, F = 20s and H = 17s, the case with tasks A, B, F, H is supposed to be executed after (A+B+F+H) 197 seconds. How much close we will be to real time (213s) depends on quantity (and quality) of measured data, total number of possible different tasks (size of problem space) and nature of process. If a process is predictable (for example machine) then result will be close to real values. If a process is not predictable (mainly manual labor) then the result will be far from real values, but it can be still usable. If a process is exact, it is possible to solve the problem with linear equations but this is not common for most situations. Even if there is a machine with quite predictable speed, its swiftness could be dependent on some case attributes (which are or are not available) and even if not, there will be still some little variance – for example machine must be supplied by material, however, the material flow can differ from machine speed. Analyzing unmeasured dynamic processes, which is also highly dependent on attributes, is not an easy issue to solve. First, we will describe how to analyze process, which is not dependent on attributes.

We have created quite simple algorithm, which was able to compute satisfactory result quite fast. Our algorithm is an iterative computation based on heuristics. Initially, we generate candidate solution to heavily speed up iteration. We make the following computation:

Input: Dataset, which contains all process instances
Output: List of predicted times for all tasks

1. **For** each record (process instance I, which contains of N tasks)
2. Compute execution time for every task in case as:
3. *t = Execution time (I) / N*
4. Add the value of *t* to time collection of the task.
5. Compute average from these times for every task.

Algorithm 2: Initial Candidates

Example:
A, B, D => 210s
A, C, D => 240s
210 / 3 = 70. A = {70}, B = {70}, D = {70}
240 / 3 = 80, A = {70, 80}, C = {80}, D = {70, 80}
Averages: A = 75, B = 70, C = 80, D = 75

Second step is the Iterative Algorithm is following:

Input: Dataset, which contains all process instances
Output: Improved list of predicted times for all tasks

1. Randomly select one task *T*.
2. **For** every record r_i (process instance):
3. **If** r_i contains *T* **then continue**
   else go to next record (2).
4. Count difference between real time and predicted time as: $diff_i = real - predicted$.
5. *diff = diff / number of records containing T*
6. *task time = task time + diff * learning coefficient*
7. Compute error E as:
   $E = \Sigma diff_i$ (sum of all differences)
8. **Repeat** algorithm until error decreases.

Algorithm 3: Iterative Algorithm

We have used the value 0.2 as a learning coefficient. Learning coefficient is a common method for iterative computing, because we do not want to converge to suboptimal solution. Good example is the perceptron learning algorithm, which was inspiration for our method.

We have tested our method in manufacturing company at a manual workplace. This workplace always makes several different tasks on doors based on customer demands. Tasks are known, however, only time of whole process instance (all tasks on one door) is available. We had about 44 thousands of process instances and about 5 thousands of different tasks. This is a big number but most of them occurred only once or twice. Only about two hundreds of tasks appeared frequently.

Management previously used simple average time for every process instance regardless what tasks were included in the subprocess. Similar method was used as reference solution in previous sections, thus we have used it again. Its error was computed in the same way as it was mentioned in in step 2 of previous algorithm. The reference error is obtained as a sum of all absolute values of difference (|real – average|) of times for all records.

Error for average time was 131 seconds (average time was 354 seconds). After application of the first algorithm (computation of initial candidate), we lowered error to 84 seconds. After few seconds run of the algorithm (iteration), error was decreased to 71 seconds. Thus, initial error was 354 ± 131 and we reduced it to 354 ± 71 seconds. This is a satisfactory result and we do not suppose it could be much more improved because of natural unpredictability of manual processes.

1) Setup Time

Every process may have a setup time. That hidden task must be done for every process instance, for example the administration or reading the line information from product and etc. It may be better to use the normative information (measured by standardizer). The problem is that the algorithm itself will provide the same error results regardless what the set up time is (because the setup time affects all the process instances in the same way, this is the problem for static processes as we have described it at the beginning of this section). But if you want to use task times not only for prediction then it is better to measure setup time by standardizer and add it to every process instance and then compute more realistic task times.

2) Average Time for Task

An intuitive way is assigning all tasks an average value. This can be done by dividing all process instances by number of tasks in it (if there is setup time, we must subtract it). Then, we will count average value from these numbers. The prediction is now counted as: Total time = setup time + number of tasks * average time for task. However, result of this method was 145 seconds, which is even worse than our base error (131). This is the reason why we did not include this method in overview. But it could work for problems with similar task times, which was not our case.

3) Genetic Algorithm

We have also tested a genetic algorithm with one point crossover, tournament selection, mutation and 50 individuals in generation. Every individual had a vector that represents times of particular tasks. Result of the genetic algorithm was not better than the previous method (also not worse), but we were testing it repeatedly for several hours. Previous algorithm was able to compute it in about half minute. Thus, our method is good enough and very fast, with results comparable to genetic algorithms.

We have tried the same genetic algorithm with starting individuals close to solution that was found by previous iterative algorithm, but result was only slightly better (error value was 69 seconds).

4) K-Nearest Neighbor

K-nearest neighbor is the good classifier as we used it previously with satisfactory results. Our result was also satisfying. It reduced error to 73 seconds (from 131s), which is very close to our previous results. However, it has a high computational demand, which can cause problem while using it in real time monitoring.

Because of different lengths of vectors (process may contain any number of tasks), there are multiple ways how to implement similarity. We have chosen this one:

At first, all close vectors are found. Those vectors must have the same length and must not differ less than in one task. But if the vector length is smaller than 3, the vectors have to be completely similar. We have got a result - collection of pair – time (that belong to vector) and weight. The similar vectors have weight equal to 1; slightly different vectors have weight equal to 0.7.

If the result collection has too few items (e.g., less than 10), it is used average time of all records instead. This last rule highly depends on nature of the process. In highly predictable process, variance is too low, thus it is possible to compute result from a small number of examples. But processes with relatively high variance will need more than few items.

We have tried several different settings of this method, but all settings led to similar results. It seems that error about 70 seconds is close to global optimal solution of the problem.

5) Process with Attributes

To gain better result, we need more information. Process attributes are suitable candidates. Process attributes can be some descriptive attributes of the process or resources involved in the process. Attributes can affect execution time of the process in the same way as affected it in previous sections. The only difference is that we also have different set of tasks in different process instances, thus ordinary classification methods are going to fail, but not while combining them with some previously mentioned methods (classification or association rules). Let us suppose, we have computed average time of every task (with some error, of course). Predicted time of process instance is a sum of average time of its tasks, as we described earlier. So we have two times – real time and predicted time. We need to compute their ratio.

$$ratio = real\ time / predicted\ timed \quad (4)$$

The ratio is now the target attribute to predict. Now, we can use any classifier (or association rules), as we described them in Sections V and VI. In the testing phase, we can obtain ratio of the process instance based on attributes. Then, we will compute predicted time based on sum of average times of its tasks. Finally, we will multiply ratio with obtained time.

We have tried the decision tree forest and difference error of 66 seconds (result of prediction without attributes is 71 seconds) was obtained. That is not an excellent result but it is usable. Problem consists in the fact that the process of computing average times of task is itself inaccurate, thus the result of classification cannot be so accurate as well. But the total result – error of 66 seconds from initial 131 seconds is still a useful result.

6) Multiple resources in process

If there is only one resource, it is possible to use resource as ordinary attribute. But what if there were multiple resources working on different tasks? Or, what if it is known how effective the resource is? We can use similar approach that was described in Sections V and VI, only with one difference. We cannot multiply classifier result with the total

process time, but we must first multiply resource efficiency with its task time. For example, John did Task A and B and Jane did task D. Times are A = 2 min, B = 3 min, D = 1 min and worker efficiency is John = 0.7 Jane = 1.1. Then total time = Classifier ratio * (2 * 0.7 + 3 * 0.7 + 1 * 1.1).

It could be quite hard to compute workers efficiency from that kind of data, thus it could be better solution to gain it differently (from some data with more precise information or from a standardizer). Of course, worker could be more skilled in one process and less skilled in another, but manager's experience tells (for manual jobs without high qualification), that if worker is slow in process A, it will be probably slow in process B as well.

7) Computing Variance

As it was said previously, variance is very important for planning– mainly for material flow and overall planning. High variance causes high inventory and there is a problem with synchronization, which can cause deviation from desired plan. Variance could by computed using this approach. The main idea behind it is that we have set of process instances (records) that contain that information – set of tasks, and real time of process execution and finally predicted time (previously computed by some of our methods). Variance for a task is computed as follows. At first, all records containing these tasks are selected.

Then, we need an average value and a current value, which is a need for variance computation. We do not know what the average value is, but we can suppose that predicted value of that record is close to average value of the process instance because this is what our methods needed to compute. Then, current value is the real value and (real time – predicted time)2 is a base assumption.

Input: Dataset containing all process instances.
Output: Values of difference for each task.

1. **For** each task *T*
2. **For** each record *r* containing *T*
3. $diff_T = diff_T + (real - predicted)^2$
4. $diff_T = diff_T$ / number of records containing T

Algorithm 4: Counting the values of difference

Note that variance computation is only an estimation, not an exact mathematical calculation. If a task causes high deviation, then the records of the task will also cause high deviation (difference between real and predicted time). However, if a task with high deviation is present many times with task with low deviation, their deviations will be average, so this is only the estimation, which is possible to be wrong and it highly depends on tasks, which are in the process instances.

8) Summary of Methods

Figure 5 is an overview of used methods. We can see that precision of all methods are nearly similar, except that genetic algorithm and K-Nearest-Neighbor takes much computing time – genetic algorithm has very long learning time and KNN very long testing time. We can also see that classification using attributes slightly improved error rate.

9) Validation on test data

We did also validation on test data (Data Set was divided by 99% for train data, 1% for test validation data). This unbalanced distribution was caused by fact that every day some new operation that was not in previous data occurred. So if we divided our data for example by 80% and 20%, results were very bad. There is a high need to learn system continuously in real production to better estimate time of brand new tasks. This is possible, because our algorithm needs about one minute to run.

We tested 99% of train data with fifty rounds of cross validation. Result was slightly worse than error in training data (about 1-3 minute to error of every method) so we did not include it here.

But we also include one little thing – in real prediction, every day a new operation with unknown time occurred. We have assigned it average time of all known tasks times.

B. General Dynamic Processes

General dynamic processes could include parallelism and every process instance could be a different process with different tasks. Resolving task performance times for general processes is more complicated than for sequential processes.

Example of some process instance with parallelism (symbol || means parallelism, symbol + serial execution): A + B + ((C + D) || (E + F)) + J.

This means that A and B are executed serially, then two parallel branches are executed – first C and then D, second E and then F and after waiting for both branches task J follows. Another example of process instance could look like that: A + (C || (E + F + B)) + D + D.

We did not deal with that kind of problem in our manufacturing company; however, we would like to propose some ideas for future research in this area. Because general dynamic process could be any process, every process instance has to contain process definition (or log with parallelism – see below), not only tasks. If only tasks are available, we may use Process Discovery first [6]. This could work in some cases, but we will first focus on processes with known model.

Let us suppose we know average times of all tasks. Then, the question is, how to compute final process instance time? Two examples above show a task execution log enhanced with parallel execution. What is the difference between normal log and log with parallelism? Normal log for both examples should look like: A, B, C, E, D, F, J and A, C, E, F, B, D, D.

This log contains information about executed tasks in order to their completion time (or start time). Using this, we do not know exactly, which tasks were executed in parallel and which of them sequentially, thus we cannot solve anything. If there is only a sequential log, process model must be available (either for every process instance or globally for all process instances). What is difference between process model and log with parallelism? Look at the second example: A + (C || (E + F + B)) + D + D.

This process instance should be executed by many different process models – for example, two final tasks D could be in process instance log, where the task D could be

in a loop that allows repetitive execution or it could be always two serial tasks D (for whatever reason).

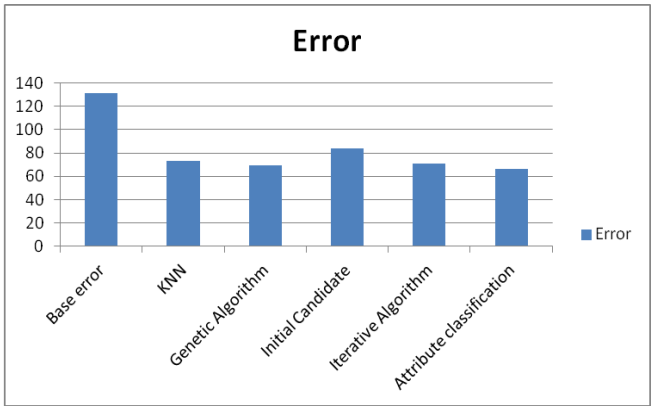


Figure 5. Results of different methods for dynamic sequential processes. Base error is computed as an error if we use average mean value from whole data set for all predictions.

How can we compute total estimated process instance time if we know average times for all tasks? If we have a log with parallelism, it is not so difficult. We must simulate this dynamic process as it is – if there is a sequential execution, we will be adding performance times. If there is a parallel execution, we will be able to continue after slower branch is completed. For example, first process (A = 2 min, B = 3 min, C = 4 min, D = 1 min, E = 1 min, F = 2 min, J = 5 min): A + B + ((C + D) || (E + F)) + J.Total = 2 + 3 + Max(4 + 1, 1 + 2) + 5 = 15 min.

Function Max returns maximum from two input numbers. If there is a more complicated process, we must use recursion. But what if we know a process model and a sequential log? We can use log replay as it is described in [6] in the section about Conformance Checking.

1) How to find solution?

How to find average times for huge set of process instances? It is important that every process instance is at least a bit different as we discussed it earlier. Static processes are always the same and there are low chances to analyze what is inside the process.

Because of the complexity of the problem, we suggest to use Genetic Algorithm. Solution can be coded as a vector of real numbers. For example, process instances could contain four tasks – A, B, C, D. Thus, we will have vector with four real numbers. Fitness of the solution is an evaluation of all process instances and computing error. Error should be computed in the same way as in Sequential Dynamic Processes – error = | real time – predicted time|. After that, genetic algorithm setting continues (selection, mutation, crossover. number of individuals in generation, static / steady state, etc.). We cannot say what setting will be the best because it highly depends on current process instances.

We believe that the genetic algorithm should be able to find average times of tasks to find suboptimal solution. But closer experiments are beyond the scope of this paper.

2) Problem with Process Discovery

Process Discovery is able to find process model from logs [6][7]. Problem of this solution is that it does not distinguish between serial execution with arbitrary order and parallel execution. For example, let us have two logs: A, B and B, A.

Most Process Discovery algorithms (as Alpha Algorithm [7]) see it as a parallel execution, because it does not depend on the order. This should be acceptable when we are analysing log to discover some usable process model that represents some probably possible executions of process, but it is not suitable for our time prediction. Sequential and parallel executions are evaluated differently. If A is 2 minutes and B is 3 minutes, than sequential execution takes 5 minutes and parallel 3 minutes. Administrative processes usually also allow this parallel execution really in parallel (if electronic documents are used). In manufactory, products cannot be produced on both in task A and task B in the same time, If parallelism is discovered then it only means that it does not depend on order of task. But there is another problem – what if there is some material in process that will be mounted on product later during processing? Now, it is possible that there is a parallel work, which indicates a problem.

However, Process Discovery could still be usable if we know which resource executed which task. If task A and B are executed by one resource, we know that even if Process Discovery says it is parallel execution, one resource must execute it sequentially.

Note that Process Mining discovers global process model for all process instances with OR routing branches, so log replay must be used to merge process model and log as we mentioned above, see Conformance Checking [6].

VIII. ANALYSING ERRORS

Primary analysis of execution time is the main focus of this paper but we can also analyse errors. Errors can be also dependent on attributes. Errors are an important part of simulation. We cannot simulate errors with very low probability but we must simulate operational errors like defective products. Error situations must be sometimes analysed separately from the time analysis (error in products must not be involved in time computation), because errors are mostly treated differently – for example defective products will be sent to repair (another task, similarity with Decision Analysis (Section III), or thrown away.

1) Classification methods

Classification methods can be easily transformed to predict errors instead of time. Error will be used as a target attribute of record for classification (0 – no error, 1 – error). After that, classifier can predict error for given attributes – for example it returns value 0.07, it means there is 7% probability of error in that task.

A. Association methods

It is also possible to use association rule based classification to predict errors. There are two classification categories in this task – 0 (“no errors”) and 1 (“error”). It is not difficult to obtain some association rules and classify the

cases but the main problem is the fact that the training data for the "error" category is much smaller than the "no errors" category because it is expected that most of products will be made without errors and no cases will be classified to "error" category.

Therefore, we will concentrate only on association rules obtained for the "error" category. Deeper analysis of this set of association rules can show us some interesting properties of cases, which lead to some error.

To use association rules for classification, some complex post-processing of association rules should be designed and implemented. This is one of the issues to be solved in the future research.

B. Unmeasured process

Error could happen in unmeasured processes. There could be many settings of how error should be handled. Process should stop immediately when error occurred, or it could run to its end and then all errors are resolved. Other information is related to error itself – if we know, which task contained error and which one did not – in this situation we know only that the process instance finished in error state. First situation will result in more precise probabilities and it is also quite simple to compute – if there are no attribute dependencies, we could only compute successful and unsuccessful (error) execution and compute ratio.

Second situation is more complicated. We know that error occurred somewhere in process, but we do not know where it is exactly located (in which tasks). However, it could be easily solved by a genetic algorithm. Vector of tasks error probabilities is a possible solution. Fitness of this solution could be computed this way. We will resolve all process instances using random generator – process instance will be replayed and error will be randomly generated using error probabilities for tasks. For example:

Assume Serial Process: ABC – A – 0.02, B – 0.01, C – 0.04. Thus, the random generator will generate error with probability 0.02 (Task A). If an error occurs then process instance will stop and it will end with error. If there was an error in historic data then fitness will be increased by 1, otherwise by zero. Vice versa, if result of random generator is successful, run of process and historic data also ends successfully, fitness will also be increased by 1. This must be repeatedly executed (at least 10-40 times, the more times, the more precise result, but more computational cost).

The result of the genetic algorithm is most probable task error rates. We did not have data for this type of problem. We included this method as another future research idea.

IX. COLLECTING DATA

We are also trying to start discussion with specialists on measurement, because measuring devices are quite expensive. There is need to join information from measurement, data mining and manufacture planning, because there is no need to measure everything. For example Half Measurement (Section III) is good example of saving money. We will now describe what is important in measurement to build simulator:

- Measure every critical task (bottlenecks). Task that are not in any critical path, don not have to be measured.
- It is good to have information about error in data, because it is problem to get it from them using only time information (Section III).
- Use Half Measurement everywhere where it is possible. Be careful, because only workplaces with low variation of production rate and high utilization are candidates for Half Measurement.
- Information about breaks is important, because we do not want to include them in production time.
- Sometimes, removable measure devices should be used to measure more tasks in different times – but be assured that nothing important has change since last measurement. Information must be valid, not obsolete.
- Machine with constant production speed, which are also independent on product attributes do not have to be measured. But be careful, sometimes preparation of product for the constant machine is dependent on product attributes, in that case, measure preparation instead of production.

X. CONCLUSION

In the paper, it has been shown that the quality of results does not depend only on our methods, but mainly on manufactory itself. For example, if execution time cannot be predicted from case attributes in expected precision, prediction will be useless. But this does not mean that the whole task is not predictable. Some tasks has little variance itself, so no advanced methods are needed.

In our company, predictions helped lower execution time variance, which is very useful in internal logistics planning, but there is a question what precision is needed to implement some better planning techniques that will enable significant saving especially in space and time needed for manufacturing production by improving input data for planning algorithms. We can also find a subset of case parameters that have low time deviation and try to optimize their production. Other cases could be produced in another time or in other machines in parallel with another approach (slower but more robust).

Resources working speed was also the big issue. In addition, dynamic aspect of process (new machines, resource improvement) is a problem to solve. We also tried other approaches like Association Rules and use them with success, but still, some problems are awaiting us, mainly because of unmeasured or partly measured processes. We introduced some solutions that worked and we hope that other problems will be solved too. This could be topic of our further research. We believe these methods could reach maturity and will be used in some manufactories in future.

ACKNOWLEDGMENT

This research was supported by the grants of MPO Czech Republic TIP FR-TI3 039, the grant FIT-S-10-2, the research plan no. MSM0021630528 and the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

REFERENCES

[1] M. Pospíšil, V. Mates, and T. Hruška, "Process Mining in Manufacturing Company," in The Fifth International Conference on Information, Process, and Knowledge Management, Nice, France, IARIA, 2013, pp. 143-148, ISBN 978-1-61208-254-7

[2] W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek, "Business process mining: An industrial application," Information Systems, Volume 32, Issue 5, July 2007, pp. 713-732, ISSN 0306-4379, DOI: 10.1016/j.is.2006.05.003.

[3] M Song and W.M.P. van der Aalst, "Towards comprehensive support for organizational mining," Decision Support Systems, Volume 46, Issue 1, December 2008, pp. 300-317, ISSN 0167-9236, DOI: 10.1016/j.dss.2008.07.002.

[4] W. M. P. van der Aalst, and A. J. M. M. Weijters, "Process mining: a research agenda", Computers in Industry, Volume 53, Issue 3, Process / Workflow Mining, April 2004, pp. 231-244, ISSN 0166-3615, DOI: 10.1016/j.compind.2003.10.001.

[5] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters, "Workflow mining: A survey of issues and approaches," Data & Knowledge Engineering, Volume 47, Issue 2, November 2003, pp. 237-267, ISSN 0169-023X, DOI: 10.1016/S0169-023X(03)00066-1.

[6] W. M. P. van der Aalst, "Process Mining," Berlin, Heidelberg 2011, ISBN 978-3-642-19344-6

[7] A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst, "Discovering simulation models," Information Systems, Volume 34, Issue 3, May 2009, pp. 305-327, ISSN 030

[8] A. Rozinat, M.T. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C.J. Fidge, "Workflow simulation for operational decision support", Data & Knowledge Engineering, Volume 68, Issue 9, Sixth International Conference on Business Process Management (BPM 2008) - Five selected and extended papers, September 2009, pp. 834-850, ISSN 0169-023X, DOI: 10.1016/j.datak.2009.02.014.

[9] W.M.P. van der Aalst, M.H. Schonenberg, and M. Song, "Time prediction based on process mining", Information Systems, Volume 36, Issue 2, Special Issue: Semantic Integration of Data, Multimedia, and Services, April 2011, pp. 450-475, ISSN 0306-4379, DOI: 10.1016/j.is.2010.09.001.

[10] M. Pospisil., T. Hruška, "Business Process Simulation for Predictions," in BUSTECH 2012: The Second International Conference on Business Intelligence and Technology, Nice, France, IARIA, 2012, pp. 14-18, ISBN 978-1-61208-223-3

[11] J. Nakatumba, A. Rozinat, and N. Russell, "Business Process Simulation: How to get it right," Springer-Verlag, 2010, doi=10.1.1.151.834

[12] J. Nakatumba and W.M.P.V.D. Aalst, "Analyzing Resource Behavior Using Process Mining", in Proc. Business Process Management Workshops, 2009, pp. 69-80.

[13] M. Pospíšil, V. Mates, T. Hruška, "Analysing Resource Performance and its Application in Company," in The Fifth International Conference on Information, Process, and Knowledge Management, Nice, France, IARIA, 2013, pp. 149-154, ISBN 978-1-61208-254-7

[14] W.M.P. Van der Aalst, "Business Process Simulation Revisited," 2010, ISSN: 1865-1348

[15] B. Wetzstein, P. Leitner, F. Rosenberg, I. Brandic, S. Dustdar, and F. Leymann, "Monitoring and Analyzing Influential Factors of Business Process Performance," Enterprise Distributed Object Computing Conference, 2009. EDOC '09. IEEE International, pp. 141-150, 1-4 Sept. 2009, doi: 10.1109/EDOC.2009.18

[16] D. Grigori, F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.C. Shan, "Business Process Intelligence, Computers," in Industry, Volume 53, Issue 3, Process / Workflow Mining, April 2004, pp. 321-343, ISSN 0166-3615, DOI: 10.1016/j.compind.2003.10.007.

[17] D. Grigori, F. Casati, U. Dayal, and M.C. Shan, "Improving Business Process Quality through Exception Understanding,Prediction, and Prevention," in Proceedings of the 27th VLDB Conference,Roma, Italy, 2001, 1-55860-804-4

[18] R. Agrawal, T. Imielinski., A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, USA, 1993, pp. 207-216.

[19] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, USA, 1994, pp. 487—499.

[20] J. Han, J, J. Pei, Y. Yin, "Mining Frequent Patterns without Candidate," Proceedings of the ACM-SIGMOD Conference on Management of Data (SIGMOD'00), Dallas, TX, 2000, pp. 1-12.

[21] B. Liu, W. Hsu, and Y. Ma, "Integrating Classification and Association Rule Mining," in ACM Conference on Knowledge Discovery and Data Mining (SIGKDD'98), New York, August 1998, pp. 80–86.

[22] W. Li, J. Han, and J. Pei.: CMAR, "Accurate and efficient classification based on multiple class-association rules," in IEEE International Conference on Data Mining (ICDM'01), San Jose, California, 2001, pp. 369 – 376.

[23] M. L. Antonie, and O. Zaiane, "Text Document Categorization by Term Association," in Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, 2002, pp. 19-26.

[24] V. Bartík, "Association Based Classification for Relational Data and Its Use in Web Mining," in IEEE Symposium on Computational Intelligence and Data Mining, Nashville, USA, 2009, pp. 252-258.

A New Representation of WordNet® using Graph Databases On-Disk and In-Memory

Khaled Nagi
Dept. of Computer and Systems Engineering
Faculty of Engineering, Alexandria University
Alexandria, Egypt
khaled.nagi@alexu.edu.eg

***Abstract*— WordNet® is one of the most important resources in computation linguistics. The semantically related database of English terms is widely used in text analysis and retrieval domains, which constitute typical features, employed by social networks and other modern Web 2.0 applications. Under the hood, WordNet® can be seen as a sort of read-only social network relating its language terms. In our work, we implement a new storage technique for WordNet® based on graph databases. Graph databases are a major pillar of the NoSQL movement with lots of emerging products, such as Neo4j. In this extended paper, we present two new graph data models for the WordNet® dictionary. We use the emerging graph database management system Neo4j and deploy the models *on-disk* as well as *in-memory*. We analyze their performance and compare them to other traditional storage models based on native file systems and relational database management systems. With this contribution, we also validate the applicability of modern graph databases in new areas beside the typical large-scale social networks with several hundreds of millions of nodes.**

Keywords-WordNet®; semantic relationships; graph databases; storage models; Neo4j; on-disk and in-memory DBMS; performance analysis.

I. INTRODUCTION

This paper is an extension of the work done in [1], whose aim is to provide new data representation models for WordNet® based on modern NoSQL graph databases. In this paper, we implement various data *storage* models for these representations varying from in-disk models, creating in-memory virtual disk representations and using pure in-memory models. It is worth mentioning that the size of the WordNet® dictionary enables the efficient employment of these variations and offers the best benchmarking platform for applications of this moderate size.

WordNet® [2] is a large lexical database of English terms and is currently one of the most important resources in computation linguistics. Several computer disciplines, such as information retrieval, text analysis and text mining, are used to enrich modern Web 2.0 applications; typically, social networks, search engines, and global online marketplaces. These disciplines usually rely on the semantic relationships among linguistic terms. This is where WordNet® comes to action.

A parallel development over the last decade is the emergence of NoSQL databases. Certainly, they are no replacement for the relational database paradigm. However, Web 2.0 builds a rich application field for managing billions of objects that do not have the regular and repetitive pattern suitable for the relational model. One major type of NoSQL databases is the *graph database* model. Since social networks can be easily modeled as one large graph of interconnected users, they can be the killer application for graph databases with their strength in traversing and navigating through huge graphs.

However, little to no work has been done to investigate the use of graph database management systems in moderate sized databases. Of course, the database has to be relationship-rich for the implementation to make sense. In our work, we implement a new storage technique for WordNet® based on graph databases. For this purpose, we present two data models and implement them on an emerging graph database management system: Neo4j [3]. Currently, Neo4j is the leading graph database management system in terms of installations and user base. WordNet® dictionary has several characteristics that promote our proposition: *it is used in several modern Web 2.0 applications*, such as social networks; *it has a moderate size of datasets*; and *traversing the semantic relationship graph is a common use case*.

Since the modeling and benchmarking experiences of these new graph databases are not as established as in the relational database model, we implement two variations and conduct several performance experiments to analyze their behavior and compare them to the relational model.

The rest of the paper is organized as follows. Section II provides a background on WordNet® and its applications as well as a brief survey on graph database technology. Our proposed system and data models are presented in Section III. In Section IV, we describe the storage models. Section V contains the results of our performance evaluation and Section VI concludes the paper and presents a brief insight in our future work.

II. BACKGROUND

A. WordNet®

The WordNet® project began in the Princeton University Department of Psychology and is currently housed in the Department of Computer Science. WordNet® is a large lexical database of English [2]. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms

(synsets), each expressing a distinct concept. A synset contains a brief definition (gloss). Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet® labels the semantic relations. The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonym, hyponym or IS-A relation). Other semantic relations include meronym (a term which denotes part of something but which is used to refer to the whole of it), antonym (a word opposite in meaning to another), and holonym (a word that names the whole of which a given word is a part). The majority of the WordNet®'s relations connect words from the same part-of-speech (POS). Valid WordNet parts-of-speech include (noun="n", verb="v", adj="a", and adverb="r"). Currently, WordNet® comprises 117,000 synsets and 147,000 words. Today, WordNet® is considered the most important resource available to researchers in computational linguistics, text analysis, text retrieval and many related areas [4]. Several projects and associations are built around WordNet®.

The Global WordNet Association [5] is a free, public and non-commercial organization that provides a platform for discussing, sharing and connecting wordnets for all languages in the world. The Mimida project [6], developed by Maurice Gittens, is a WordNet-based mechanically generated multilingual semantic network for more than 20 languages based on dictionaries found on the Web. EuroWordNet [7] is a multilingual database with wordnets for several European languages (Dutch, Italian, Spanish, German, French, Czech and Estonian). It is constructed according to the main principles of Princeton's WordNet®. One of the main results of the European project that started in 1996 and lasted for 3 years is to link these wordnets to English WordNet® and to provide an Inter-Lingual-Index to connect the different wordnets and other ontologies [8]. MultiWordNet [9], developed by Luisa Bentivogli and others at ITC-irst, is a multilingual lexical database. In MuliWordNet, the Italian WordNet is strictly aligned with the Princeton WordNet®. Unfortunately, it comprises a small subset of the Italian language with 44,000 words and 35,400 synsets. Later on, several projects, such as ArchiWN [10], attempt to integrate WordNet with domain-specific knowledge.

RitaWN [11], developed by Daniel Howe, is an interesting library built on WordNet®. It provides simple access to the WordNet ontology for language-oriented artists. RitaWN provides semantically related alternatives for a given word and parts-of-speech (POS) such as returning all synonyms, antonyms, hyponyms for the noun "cat". The library also provides distance metrics between ontology terms, and assigns unique IDs for each word sense/pos.

Several projects aim at providing access to the WordNet® native dictionary. For example, JWNL [12] provides a low-level API to the data provided by the standard WordNet® distribution. In its core, RitaWN uses JWNL to access the native file-based WordNet® dictionary. Other projects, such as WordNetScope [13], WNSQL [14], and wordnet2sql® [15], provide a relational database storage for WordNet®.

B. Graph Databases

NoSQL databases are older than relational databases. Nevertheless, their renaissance came first with the emergence of Web 2.0 during the last decade. Their main strengths come from the need to manage extremely large volumes of data that are collected by modern social networks, search engines, global online marketplaces, etc. For this type of applications, ACID (Atomicity, Consistency, Isolation, Durability) transaction properties [16] are simply too restrictive. More relaxed models emerged such as the CAP (Consistency, Availability and Partition Tolerance) theory or eventually consistent [17], which in general means that any large scale distributed DBMS can guarantee for *two* of *three* aspects: *Consistency*, *Availability*, and *Partition tolerance*. In order to solve the conflicts of the CAP theory, the BASE consistency model (Basically, soft state, eventually consistent) was defined for modern applications [17]. In contrast to ACID, BASE concentrates on availability at the cost of consistency. BASE adopts an optimistic approach, in which consistency is seen as a transitional process that will be *eventually* reached. Together with the publication of Google's BigTable and Map/Reduce frameworks [18], dozens of NoSQL databases emerged. A good overview of existing NoSQL database management systems can be found in [19].

Mainly, NoSQL database systems fall into four categories:

- Key-value systems,
- Column-family systems,
- Document stores, and
- Graph databases.

Graph databases have a long academic tradition. Traditionally, research concentrated on providing new algorithms for storing and processing very large and distributed graphs. These research efforts helped a lot in forming object-oriented database management systems and later XML databases.

Since social networks can be easily viewed as one large graph of interconnected users, they offer graph databases the chance for a great comeback. Since then, the whole stack of database science was redefined for graph databases. At the heart of any graph database lies an efficient representation of entities and relationships between them. All graph database models have, as their formal foundation, variations on the basic mathematical definition of a graph, for example, directed or undirected graphs, labeled or unlabeled edges and nodes, hypergraphs, and hypernodes [20]. For querying and manipulating the data in the graph, a substantial work focused on the problem of querying graphs, the visual presentation of results, and graphical query languages. Old languages such as G, G++ in the 80s [21], the object-oriented Pattern Matching Language (PaMaL) in the 90s [22], through Glide [23] in 2002 appeared. G is based on regular expressions that allow simple formulation of recursive queries. PaMaL is a graphical data manipulation language that uses patterns. Glide is a graph query language where queries are expressed using a linear notation formed by

labels and wildcards. Glide uses a method called GraphGrep [23] based on sub-graph matching to answer the queries.

However, modern graph databases prefer providing traversal methods instead of declarative languages due to its simplicity and ease of use within modern languages such as Java. Taking Neo4j as example, when a `Traverser` is created, it is parameterized with two evaluators and the relationship types to traverse, with the direction to traverse each type. The evaluators are used for determining for each node in the set of candidate nodes if it should be returned or not, and if the traversal should be pruned (stopped) at this point. The nodes that are traversed by a `Traverser` are each visited exactly once, meaning that the returned iterator of nodes will never contain duplicate nodes [3].

Several systems such as Neo4j [3], InfoGrid [24], and many other products are available for research and commercial use today. Typical uses of these new graph database management systems include social networks, GIS, and XML applications. However, they did not find application in moderate sized text analysis applications or relationship mining.

III. PROPOSED SYSTEM AND DATA MODEL

Fig. 1 provides an overview of the proposed implementation. RitaWN [11] provides synonyms, antonyms, hypernyms, hyponyms, holonyms, meronyms, coordinates, similars, nominalizations, verb-groups, derived-terms glossaries, descriptions, support for pattern matching, soundex, anagrams, etc. In Fig. 1, RitaWN is represented by an arbitrary client in this domain, which sends semantic inquiries and receives the results as a list of related terms. In the actual RitaWN, the library wraps Jawbone/JWNL [12] functionality for Java processing; which, in turn, accesses the native WordNet® dictionary.

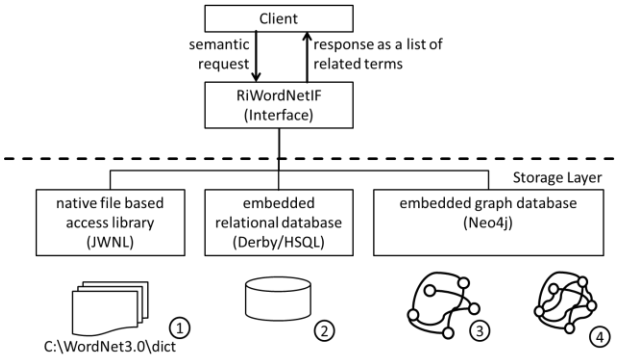


Figure 1. Architecture of the proposed system.

In order to separate the data representation model from the logic, we extract a `RiWordNetIF` Java interface. The interface defines methods to return semantically related words. The methods are categorized into 4 groups in ascending complexity with respect to reaching the returned values:

- Attribute inquiries: these methods return single attribute values for a given word, such as `String getBestPos(String w)` and `boolean isNoun(String w)`.
- Semantic relationships inquiries: in this set, methods return all semantically related words for a given word and POS, such as `String[] getHolonyms(String w, String pos)` and `String[] getHypernyms(String w, String pos)`. In our system, we define eight such methods.
- Relationship tree inquiries: in this set of methods, the library returns the whole path from the first synset for a given word and POS to the root word. Typical root words in WordNet® are "Entity" or "Object". In our implementation, we have `String[] getHyponymTree(String w, String pos)` and `String[] getHypernymTree(String w, String pos);` which basically trace back `getHyponym(String w, String pos)` and `getHypernym(String w, String pos)` respectively to the root word.
- Common parent inquiries: methods of this group find a common semantic path between two words in a POS subnet by traversing the WordNet® synset graph. For example, the method `String[] getCommonParent(String w1, String pos, String w2)` finds the following path illustrated in Fig. 2 for the nouns "dog" and "animal". Traversal is done based on a Depth First Search algorithm with a slight adaptation to stop traversing whenever one of the synsets of the sink term `w2` is reached.

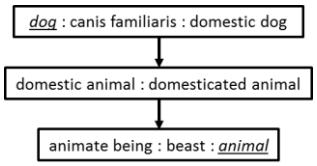


Figure 2. Semantic path from 'dog' to 'animal'.

A. Data Model

In the storage layer, illustrated in the lower part of Fig. 1, we provide four different representations for the WordNet® dictionary as described in the following subsections.

1) File-based Model

In its original implementation, RiTa.WordNet uses the JWNL [12] library to directly browse the native dictionary provided by a standard WordNet® installation. As will be shown later, this implementation has the worst performance. We use it for validation purposes for the other three implementations.

2) Relational Database Model

We use a database model similar to the one used in [15]. Fig. 3 illustrates a UML class diagram for the relevant

classes. The `words` entity has a `wordid` as a primary key, the `lemma` definition and the different `POSs` are coded as string with the best POS as the first character of the string. Similarly, the `synsets` entity holds all WordNet® synsets, their `POS`, and definition. The primary key is `synsetid`. The many-to-many relationship between words and synsets is modeled by the `senses` entity. It contains the foreign keys `wordid` and `synsetid`. `Synsets` are related to each other via the `semlinks` entity. `Synset1id` points to the `from` direction and `Synset2id` to the `to` direction. The types of semantic links are defined by `linkid` which is a foreign key to the `linktype` entity. All types of links are listed in the `linktype` entity.

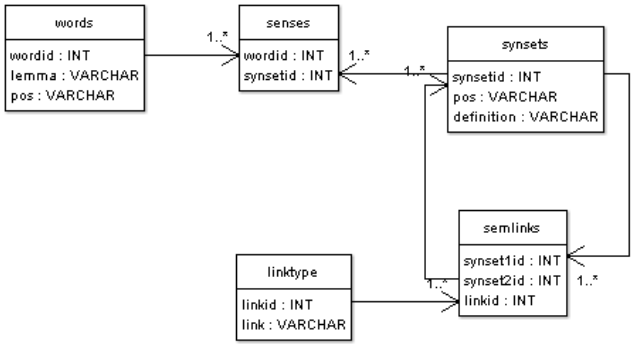


Figure 3. UML class diagram for the relational database.

3) Graph Database Model

In our proposed work, we model the WordNet® as a graph database. An object diagram is illustrated in Fig. 4. We have two types of nodes: `words` (illustrated as ellipses) and `synsets` (illustrated as hexagons). The attributes of a word are a lemma and the different `POSs`, which are coded as a string with the best POS as the first character of the string. The `synset` has a property `definition`. There exists a bi-directional relation `Rel_sense` between `words` and `synsets`. The attribute `pos` of the relation indicates the POS associated with the sense. `Synsets` are interconnected by directed relations. These relationships `Rel_SemanticLink` carry the `type` of the link in the attribute type. For example, in Fig. 4, word *w1* has one sense as a noun with link to `sysnset` *sa* and two senses as verbs for `synsets` *sc* and *sd*. `Synset` *sa* has two hyponyms *sb* and *se* by following the relationships `Rel_SemanticLink` with type "hyponym". *w4* has one sense *sb* as a noun. *w2* and *w3* – as nouns - share the same `synset` *se*. *w5* has only one sense as a verb which is *sc*. So, if `getHoponyms("w1", "n")` is called, the result will be *w2*, *w3*, and *w4*.

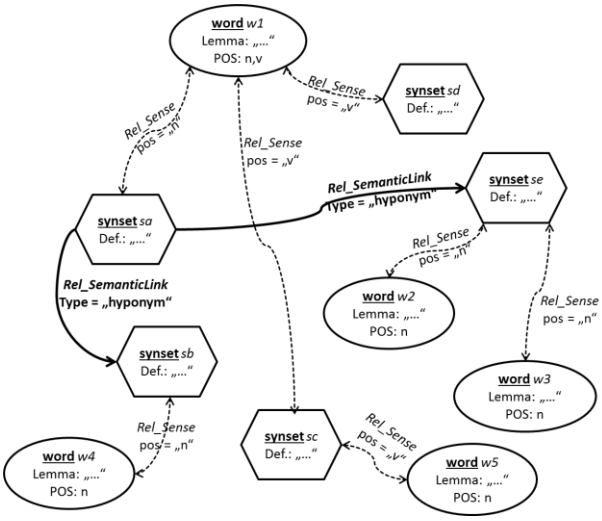


Figure 4. Object diagram for the proposed WordNet® graph database model.

4) Graph Database Storage with Additional Directly Derived Relationships

In the RiTa.WordNet application scenario, we expect many inquiries about semantically related words (e.g., hyponyms, synonyms, meronyms, etc.). Synsets are mainly the means to return the semantically related words. At the same time, the application is typically read-only and represents a good example for a wide range of read-only (or low-update/high-read) applications. The graph database is only updated with the release of a new WordNet® dictionary. This motivates us to augment the design mentioned in the previous section with the derived semantic relationships between words and not only synsets. The idea is similar to *materialized* views known in relational databases. The result of semantic relationship inquiries (e.g., `getHyponyms()`, `getSynonyms()`, `getMeronyms()`, etc.) is generated by traversing only one relationship for each result word. We intuitively expect a quicker response time at the cost of a high storage volume since the connectivity of the graph is highly increased. In the case of the limitation of the client application to inquiries within the above-mentioned four categories, the original relationships can be even dropped.

In terms of implementation, these relationships are identified through the relationship type. Fig. 5 illustrates the derived relationships for the example in Fig. 4. Only the relationship of type `Rel_Hyponym` for noun POS of word *w1*; namely, *w2*, *w3*, and *w4* is drawn. For more complex inquiries outside the categories "relationship tree" and "common parent", a combination of original and derived relationships are used in the traversal.

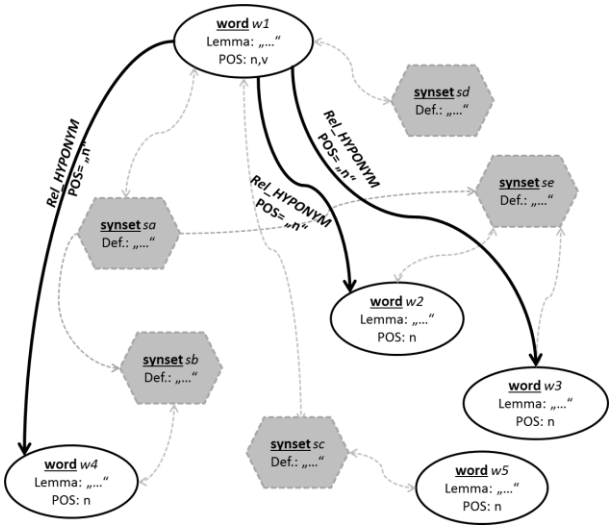


Figure 5. Object diagram with the extra derived relationships.

IV. STORAGE MODEL

We implement the graph data models using the currently leading graph database management system: Neo4j [3]. For all implementation models, we attempt to store the data *on-disk*. In addition to our work done and presented in [1], we also provide implementations stored *in-memory*.

A. On-Disk implementations

Using on-disk implementations is the traditional way for storing data. It preserves the content after system shutdown but suffers from the latency of hard disks.

In the file-based data model, WordNet® data is stored within the WordNet installation directory on disk. Native access is done through JWNL [12] library.

As for the relational database model, we choose Apache Derby [25] as the database management system to hold this data model. Apache Derby is part of the Apache Group. It gained a good reputation and a high spread for applications requiring *embedded* relational DBMS. We explicitly rule out the usage of larger relational database management systems running in server mode, such as Oracle or DB2, since we are concerned with the use case of relatively small-sized read-only interrelated data sets. Apache Derby is distributed as a Java jar file to be added to the classpath of the application. It also comes as a stand-alone version. In this case, the data resides in the database container on disk. We follow the common practices for standard relational database by building indices on the primary and foreign keys.

For the two data models we introduce in our research, we provide implementations for the emerging graph database management system Neo4j [3].

From its background and growing customer base, it is clear that Neo4j enjoys an increasing wide spread especially in the industry. Another advantage over InfoGrid [24] is its ease of use as it does not require the explicit definition of the model of the schema in XML as in the case of InfoGrid, which renders the addition of more entity types to the graph more simple. The basic setup for Neo4j is that the data is stored in a proprietary format on-disk. Neo4j then provides various data caching strategies in memory for so-called hot-spot data access.

B. In-Memory implementations

For the in-memory implementations, the whole WordNet® content is loaded in memory from the permanent storage during system startup. Having the content cached in memory avoids any access to the hard disk. The moderate size of the WordNet® data enables this setting.

We create a virtual disk out of RAM using RamDisk Plus [26], which uses a patented memory management component that makes a predefined portion of the RAM appear as a physical hard disk to the operating system and programs. The file-based data model of WordNet® is simply deployed on this virtual hard disk and the same JWNL [12] library is used to access the content.

In the case of the relational model, we experiment using two options:

- Similar to the file-based implementation, the Apache Derby database is stored in the virtual RAM Disk.
- We migrate the implementation to HSQL [27], which provides an in-memory transient storage mechanism for its tables. During startup, the content is loaded from the permanent storage into the in-memory tables created by the `CREATE MEMORY TABLE` SQL command.

Finally, for the two Neo4j data models, we also try the following two settings:

- Similar to the file-based and the relational implementations, we store both graph data models on the virtual RAM Disk.
- We set the cache management policy in Neo4j to `strong`. This cache setting holds on to all data that gets loaded to never release it. Additionally, Neo4j store can use memory mapped I/O for reading/writing. For optimized I/O access, Neo4j uses the `java.nio` package. Native I/O results in memory being allocated outside the normal Java heap so that memory usage needs to be taken into consideration. In order to get the best out of this setting, we increase the size of the cache used and the size of the memory mapped I/O to hold all the WordNet® data content.

V. PERFORMANCE EVALUATION

In order to evaluate the performance of our proposed system, we provide *four* implementations for the Java interface `RiWordNetIF` mentioned in Section III. The implementations are file-based storage, relational DBMS using Apache Derby and HSQL, the graph database using Neo4j, and a second implementation using the materialized directly derived relationships also using Neo4j. For each one of the settings, we deploy the implementation twice: *on-disk* and *in-memory*.

It is important to notice that the purpose of this evaluation is to give a general impression on the performance impact and not to give concrete benchmarking

figures. For sure, the optimization of all DBMS implementations; such as using indices or even exchanging the DBMS itself versus using future versions of Neo4j might lead to different results. *We would be satisfied if our proposed solution provides slightly better results than relational DBMS.* It is interesting to observe the effect of using in-memory and large caching settings for the different data model strategies on a moderately sized content like WordNet® as well.

We develop a simple performance evaluation toolkit around our implementations. A workload generator sends inquiries to all back-ends. The inquiries are grouped into four categories, as mentioned in Section III. The workload generator submits the inquiries in parallel to the application with each inquiry executing in a separate thread.

The input for the inquiry is chosen at random from an input file containing WordNet® words and their associated best POS. In case of `getCommonParent()`, another input file is used, which contains tuples of somehow related words, together with their common POS (e.g., "tiger", "cat", and "noun"). The tuples are chosen carefully to yield paths of different lengths.

The performance of the system is monitored using a performance monitor unit that records the response time of each inquiry and the number of inquiries performed by each thread in a regular time interval.

A. Input Parameters and Performance Metrics

The number of concurrent inquiry threads is increased from 1 to 50. Each experiment executes on each back-end for 5 minutes in order to eliminate any transient effects and measure the system performance after the 'warm-up' phase. The experiments are conducted for each type of inquiries separately.

In all our experiments, we monitor the system *response time* in terms of microseconds per operation from the moment of submitting the inquiry until receiving the result.

We also monitor the system *throughput* in terms of inquires per hour for each thread.

B. System Configuration

In our experiments, we use an Intel CORE™ i7 vPro 2.7GHz processor, 8 GB RAM and a Solid State Drive (SSD). The operating system is Windows 7 64-bits. In order to build in-memory storage, we use RamDisk Plus [26].

We use JDK 1.6.0, Neo4j version 1.6 for the graph database engine, embedded Derby™ version 10.7.1.1 and HSQL version 2.3.0 for the SQL back-ends, JWNL library version 1.4 [12] for file system based storage.

C. Experiment Results

The performance evaluation considers all four types of inquiries:

- Attribute,
- Semantic relationships,
- Relationship trees, and
- Common parent

for the four back-end implementations for both *on-disk* and *in-memory* settings.

We drop plotting the results of the native file system-based implementation from our graphs, although it is the only available implementation previous to this research. The reason behind this is that the results are far worse than the other implementations. The difference in most cases is more than *one order of magnitude* as can be seen on the exemplary plot of Fig. 6 of the response time of one the experiments. We also drop plotting the results of HSQL implementation in-memory, since the deployment using the combination of Apache Derby and RamDisk Plus always supersedes the relational implementation of HSQL using its in-memory feature. In all legends of the subsequent figures, *NEO DD* means using Neo4j with the additional Directly Derived Relationships, *NEO noDD* means using Neo4j with the original relationships, and *SQL Derby* denotes the implementation using the SQL Apache Derby embedded relational database management system.

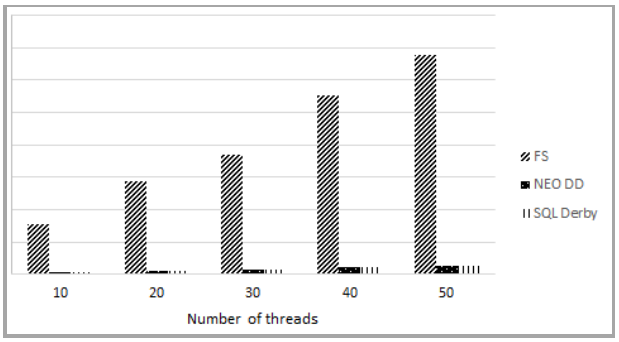


Figure 6. Average response time across increasing the number of threads with the File System (FS) included in the grpah.

1) Attribute inquiries

a) On-disk experiments

In this set of experiments, the inquiries sent by the workload generator comprise attribute inquiries only. Both response time, illustrated in Fig. 7, and throughput, illustrated in Fig. 8, degrade gracefully with the increase in number of threads while having good absolute values. Remarkably, the simple Neo4j implementation (without the extra directly derived relationships) has a 20% better response time than the other two implementations, while the full blown Neo4j implementation has a 40% decrease in system throughput. The reason for that is the attribute inquiries are mainly affected by the node (or tuple in case of relational databases) retrieval and caching. No relationship traversal is done and hence the Neo4j only suffers from its large database size especially with the augmented directly derived relationships (see Section V.*E*).

In summary, this set of experiments demonstrates that the caching mechanisms of graph databases are in general as good as the relational databases and that simple operations without graph traversals are not underprivileged in this environment.

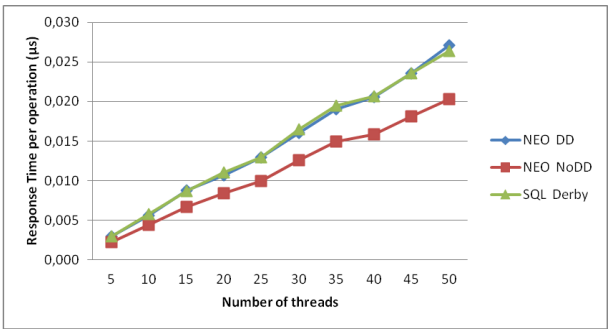


Figure 7. Response time for attribute inquiries (on-disk).

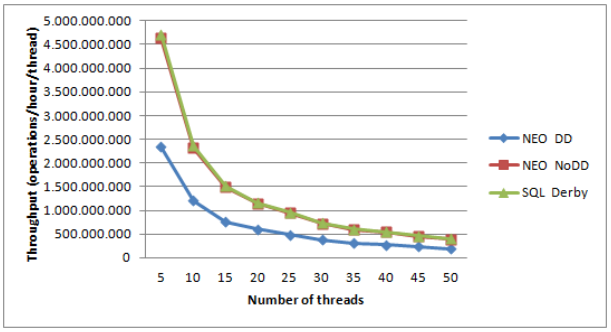


Figure 8. Throughput for attribute inquiries (on-disk).

b) In-Memory experiments

We repeat the same set of experiments using the RamDisk Plus settings explained in Section IV.*B*. The response time is plotted in Fig. 9 and the throughput for attribute inquiries in Fig. 10.

These figures indicate exactly the same behavior as their corresponding experiments in the on-disk Section. The relative decrease in response time and the relative increase in system throughput is explained separately and more elaborately in Section V.*D*.

From Fig. 9, it is clear that the response time of the simple Neo4j implementation is still the best by approx. 20%, while the throughput of the full-blown Neo4j has the worst values among the three implementations.

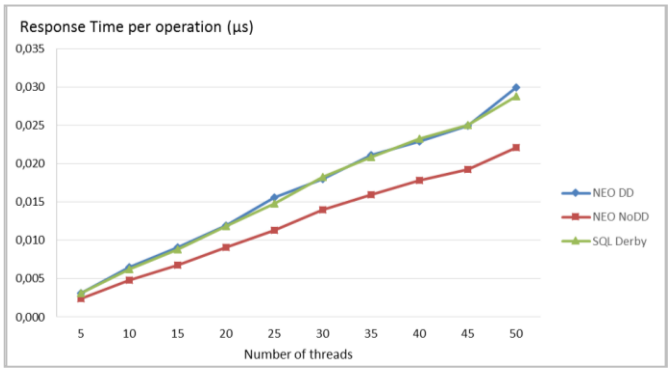


Figure 9. Response time for attribute inquiries (in-memory).

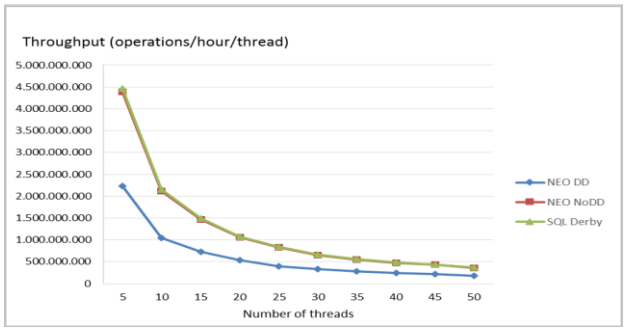


Figure 10. Throughput for attribute inquiries (in-memory).

2) Semantic relationship inquiries

a) On-disk experiments

In this set of experiments, the explicit storage of semantic relationships shows its benefit. The results are retrieved by traversing one relationship only, in contrast to 3 for the simple implementation and several joins in the relational database implementation. The response time, as illustrated in Fig. 11, is enhanced by approx. 50% for all number of threads when compared to Apache Derby and 30% by adding these directly derived relationships to a simple Neo4j implementation. However, all three back-ends behave identically when it comes to throughput as illustrated in Fig. 12. The absolute values are far below those of the simple attribute inquiries described in the previous section, which is expected due to the complexity of these inquiries as compared to attribute inquiries. In case of response time, it is almost 10 times higher than the previous set of experiments. The same applies to the throughput, which is lower by a factor of 10 as well.

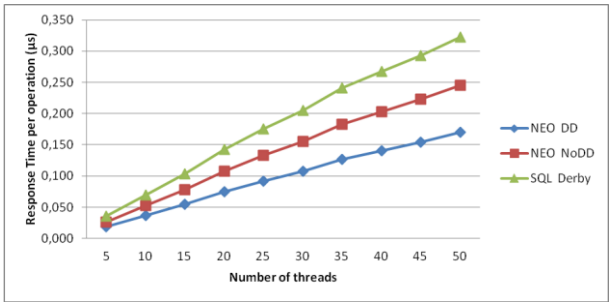


Figure 11. Response time for semantic relationship inquiries (on-disk).

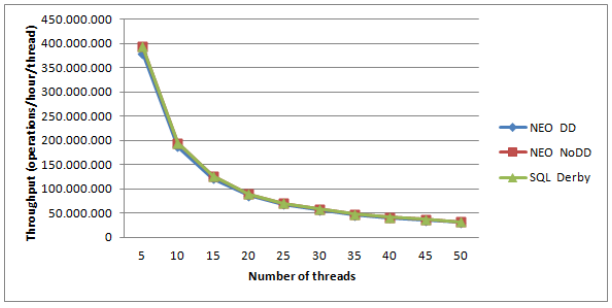


Figure 12. Throughput for semantic relationship inquiries (on-disk).

b) In-Memory experiments

The semantic relationship inquiries are repeated for the virtual disk settings. Here again, the same system behavior in terms of response time and througput is identical as the on-disk experiments. Fig. 13 illustrates the same response time pattern as in Fig. 11 and Fig. 14 illustrates that all three back-ends behave identically when it comes to throughput; which is the same scalability behavior as in the on-disk setting. The absolute values, as illustrated in Section V.*D* are almost the same as compared to Fig. 11 and Fig. 12.

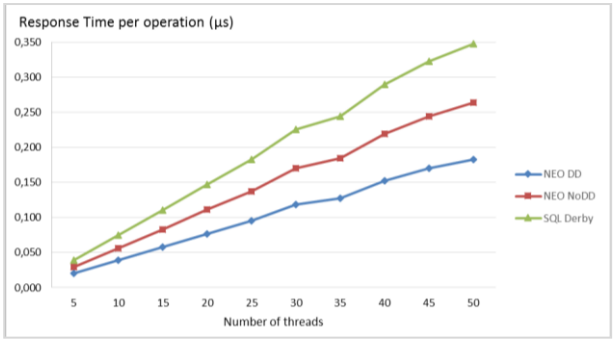


Figure 13. Response time for semantic relationship inquiries (in-memory).

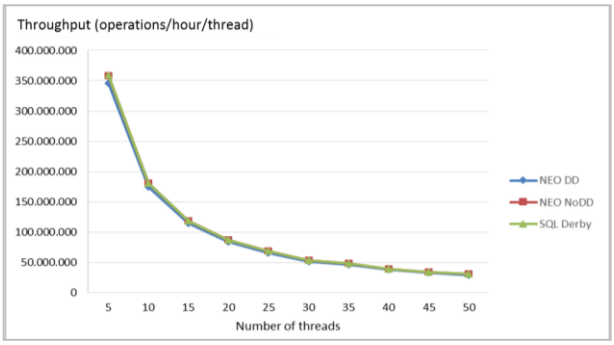


Figure 14. Throughput for semantic relationship inquiries (in-memory).

3) Relationship tree inquiries

a) On-disk experiments

The operations of this set of experiments are more complex than the previous ones. This explains the drop in absolute values of the response time and throughput, illustrated in Fig. 15 and Fig. 16, respectively when compared to the previous experiment. This time the degradation factor is only 4. Yet, the system behavior remains the same. The response time of Neo4j with the directly derived relationships is half that's of the SQL implementation. Even without the extra relationships, the response time of Neo4j is 25-30% better than the relational model. Here, again, the throughput, illustrated in Fig. 16, for all three implementations is the same. The equality of the throughput performance index of Apache Derby and the Neo4j implementations, despite the short response time of the later, is an indication that the internal pipeline capabilities of Neo4j is *not* as good as that of the relational model.

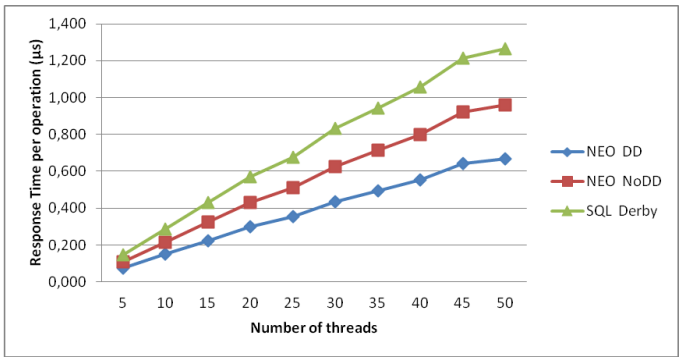


Figure 15. Response time for relationship tree inquiries (on-disk).

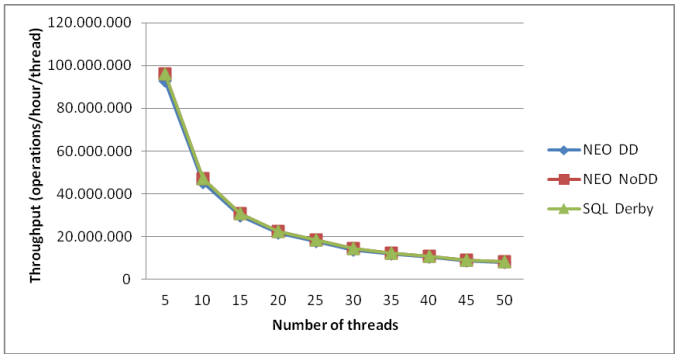


Figure 16. Throughput for relationship tree inquiries (on-disk).

b) In-Memory experiments

The same trend as the semantic relationship inquiries continues with the relationship tree inquiries when running in-memory.

The same drop in absolute values by a factor of 4 when compared to the semantic relationship inquiries is also reported here. As illustrated in Fig. 17, the response time of Neo4j with the directly derived relationships is half that's of the SQL implementation using Apache Derby.

The response time of Neo4j without the extra relationship remains in the middle of both curves. The Throughput illustrated in Fig. 18 for all implementations remains identical.

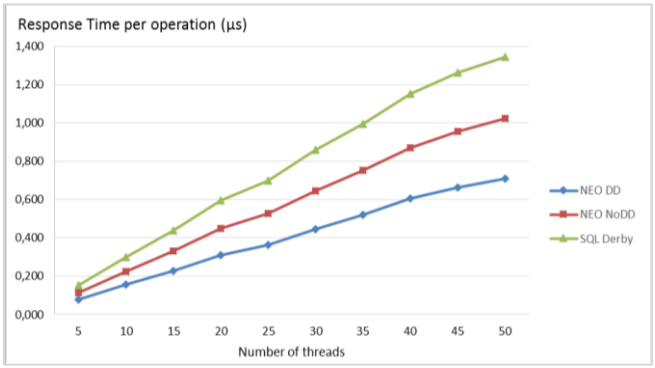


Figure 17. Response time for relationship tree inquiries (in-memory).

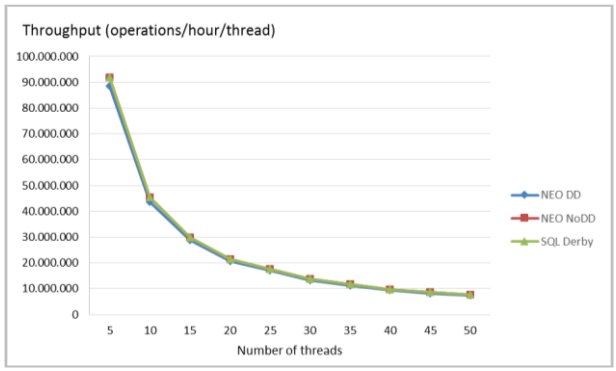


Figure 18. Throughput for relationship tree inquiries (in-memory).

4) Common parent inquiries

a) On-disk experiments

The inquiries for this set of experiments are the most complicated among all experiments. Yet, this is a very common use case in social networks. For example, in XING [28], the user can always see all paths of relationships leading from the user to any arbitrary user in the network. No wonder here that Neo4j implementations outperform the Apache Derby implementation (and the file system implementation which seems to be not able to handle all the running threads) in requesting depth first searches of the semantic network of WordNet®.

Again, Fig. 19 illustrates the extreme superiority of graph database, especially with the addition of the extra relationships. The response time is also enhanced by 45% and 30% with and without directly derived relationships, respectively.

The throughput, illustrated in Fig. 20, holds its trend across all experiments of being almost the same for the three implementations (and omitting the file system implementation of course, whose values cannot be plotted with the same scale next to their counterparts).

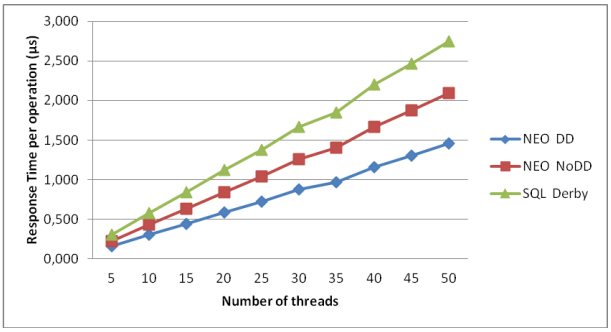


Figure 19. Response time for common parent inquiries (on-disk).

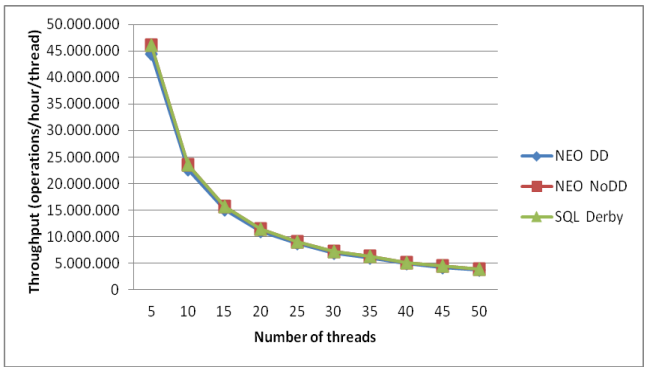


Figure 20. Throughput for common parent inquiries (on-disk).

b) In-Memory experiments

Similar to all previous in-memory experiments, the common parent inquiries yield the exact same curves as their on-disk counterparts illustrated in Fig. 21 and Fig. 22.

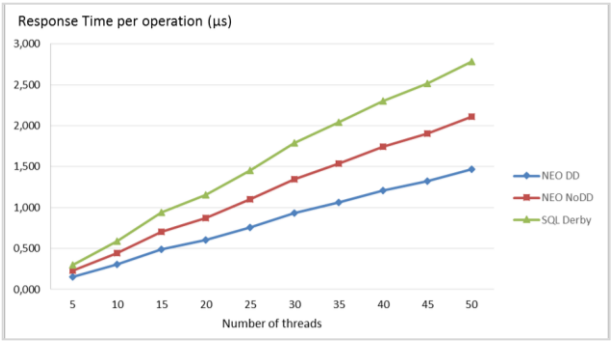


Figure 21. Response time for relationship tree inquiries (in-memory).

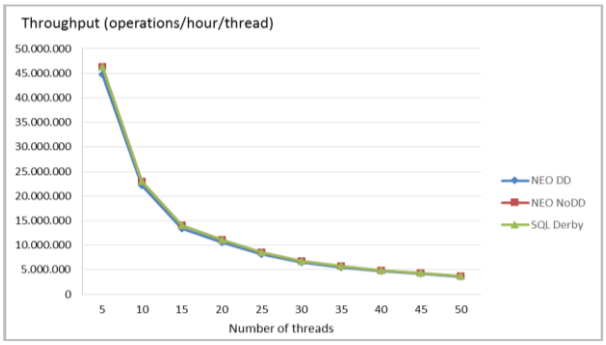


Figure 22. Throughput for relationship tree inquiries (in-memory).

D. Comparison Between On-Disk and In-Memory Performance

In this section, we compare the performance of the on-disk implementations versus their counterpart experiments done in-memory. The target is to evaluate the performance gain – if any- when keeping the whole content of WordNet® in memory. In Table I, we list the relative change in response time for each inquiry type. We define the average relative change in response time over all experiments to be:

$$relative\ change\ in\ response\ time = \frac{response\ time(in-memory) - response\ time(on-disk)}{response\ time\ (on-disk)}$$

TABLE I. CHANGES IN RESPONSE TIME HD VS. MEM

Inquiry type	FS	SQL Derby	NEO DD	NEO NoDD
Attribute	-11%	8%	11%	8%
Semantic relationships	7%	7%	7%	7%
Relationship trees	4%	5%	4%	5%
Common parent	4%	5%	4%	5%

Similarly, we list the relative change in throughput for each inquiry type in Table II. Analogously, we define the average relative change in throughput over all experiments to be:

$$relative\ change\ in\ throughput = \frac{throughput(in-memory) - throughput(on-disk)}{throughput(on-disk)}$$

TABLE II. CHANGES IN THROUGHPUT HD VS. MEM

Inquiry type	FS	SQL Derby	NEO DD	NEO NoDD
Attribute	13%	-8%	-10%	-8%
Semantic relationships	-6%	-6%	-6%	-6%
Relationship trees	-4%	5%	-5%	5%
Common parent	-4%	-4%	-4%	-4%

Remarkably, the performance does not increase substantially. In several experiments, the performance indices even slightly degrade. In all cases, the increase/decrease in performance remains within the ±10% range. This is attributed to the relatively *small size* of the WordNet® content as will be seen in the coming Section. The normal caching mechanisms provided by Apache Derby and Neo4j result in loading the whole content in-memory and renders the usage of the virtual RAM disk and all further memory optimization settings *needless*.

E. Storage Requirements

Performance in terms of good response time comes with its price. Fig. 23 illustrates the storage requirements for all four implementations. The Apache Derby and the normal Neo4j implementation occupy slightly more than double the original size of the WordNet® file-based dictionary. The redundant relationships account for more than 350 MB, making the size of the graph database 12 times larger than the file-based dictionary taken as a reference point. The good side of this particular application scenario is the absolute size of the back-ends is affordable by any desktop application. As the in-memory experiments also show, there is no need to implement extra virtual disks or extravagant caching settings, since the size of the largest implementation fits easily in the heap of any Java virtual machine of moderate size.

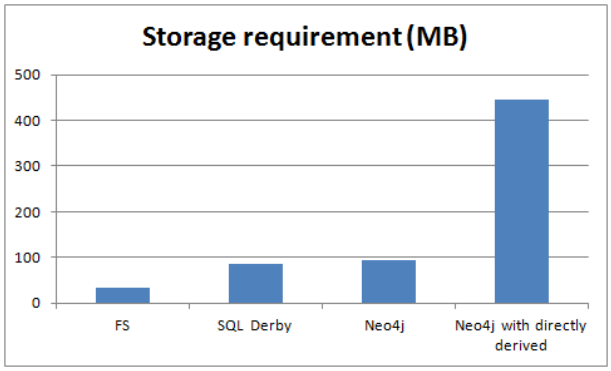


Figure 23. Storage for each back-end implementation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present two Neo4j graph data models for the WordNet® dictionary. We use Ri.WordNet as a typical client application that submits semantic inquiries discovering the relationships between English terms. We divide the inquiries into 4 categories depending on the complexity of their operations. Our performance analysis demonstrates that graph databases yield much better results than traditional relational databases in terms of response time even under extreme workloads thus speaking for their promised scalability. We also show that storing materialized directly derived relationships can improve the performance by factors of 2. This redundancy has its price in terms of storage requirements, which is acceptable due to the moderate size of the database with 117,000 synsets and 147,000 terms and the read-only nature of this small-scale social network. We also prove that there is no need for extra measures to hold the moderate size WordNet® content in memory by using in-memory databases, creating virtual RAM disks, or substantially increasing the caching mechanisms. In all our experiments, the on-disk deployments yield almost the same performance as the in-memory settings. On the long run, i.e., after having the Neo4j warm-started, almost all of the dataset is cached in memory by the underlying graph database management system. *The reason is that the WordNet® database fits in the heap of the normal Java virtual machine even with the materialization on the redundant relationships.* This adds to the advantages of using the graph databases in such moderate-sized scenarios, since the benchmarks demonstrate that there is no real need to spend extra effort in tweaking the memory usage.

One important contribution of this work is that it opens the door for new application areas for NoSQL databases (in this case the Neo4j graph database), namely smaller read-intensive database applications, in contrast to typical

applications of the NoSQL in large scale Web 2.0 such as social networks.

Yet, this is only the beginning. In the future, we plan to benchmark other graph database providers, such as InfoGrid [24]. We also plan to migrate several research done on relationship mining to work on graph database back-ends. If the benchmarking experiments show promising results, this will open the door for the application of graph databases in OLAP applications. Another extension area is the comparison against other types of NoSQL such as XML databases, document stores or column-family systems.

REFERENCES

[1] K. Nagi, "A New Representation of aWordNet® using Graph Databases," 5th International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA, Seville, 2013.

[2] C. Fellbaum, "WordNet and wordnets," in Encyclopedia of Language and Linguistics, Second Edition, Brown, Keith et al., Eds. Elsevier, Oxford, 2005, pp. 665—670.

[3] Neo4j. The World's Leading Graph Database, http://www.neo4j.org [retrieved: December, 2013].

[4] E. Voorhees, "Using WordNet for Text Retrieval," In WordNet An Electronic Lexical Database, C. Fellbaum, Ed., 0-262-06197-X. MIT Press, 1998.

[5] The Global WordNet Association, http://www.globalwordnet.org [retrieved: December, 2013].

[6] Mimida: A mechanically generated Multilingual Semantic Network, http://gittens.nl/gittens/topics/SemanticNetworks.html [retrieved: December, 2013].

[7] P. Vossen, "EuroWordNet: a multilingual database for information retrieval," DELOS workshop on Cross-language Information Retrieval, Zürich, 1997.

[8] P. Vossen, W. Peters, and J. Gonzalo, "Towards a Universal Index of Meaning," ACL-99 Siglex workshop, Maryland, 1999.

[9] E. Pianta, L. Bentivogli, and C. Girardi, "MultiWordNet: developing an aligned multilingual database," 1st International Conference on Global WordNet, Mysore, India, 2002.

[10] L. Bentivogli, A. Bocco, and E. Pianta, "ArchiWordNet: Integrating WordNet with Domain-Specific Knowledge," 2nd Global WordNet Conference, Brno, Czech Republic, 2004, pp. 39—46.

[11] RiTa.WordNet: a WordNet library for Java/Processing, http://www.rednoise.org/rita/wordnet/documentation [retrieved: December, 2013].

[12] Java WordNet Library, http://sourceforge.net/projects/jwordnet [retrieved: December, 2013].

[13] WordNetScope, http://wnscope.sourceforge.net [retrieved: December, 2013].

[14] WordNetSQL, http://wnsql.sourceforge.net [retrieved: December, 2013].

[15] wordnet2sql, http://www.semantilog.org/wn2sql.html [retrieved: December, 2013].

[16] J. Gray, and A. Reuter, "Transaction Processing: Concepts and Techniques," Morgan Kaufmann, 1983.

[17] E. Brewer, "Towards Robust Distributed Systems," ACM Symposium on Principles of Distributed Computing, Keynote speech, 2000.

[18] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, and A. Fikes, "Bigtable: A distributed storage system for structured data," 7th Symposium on Operating System Design and Implementation. Seattle, 2006.

[19] S. Edlich, A. Friedland, J. Hampe, and B. Brauer, "NoSQL: Introduction to the World of non-relational Web 2.0 Databases," (In German) NoSQL: Einstieg in die Welt nichrelationaler Web 2.0 Datenbanken. Hanser Verlag, 2010.

[20] R. Angles, and C. Gutierrez, "Survey of Graph Database Models," ACM Computing Surveys, Vol. 40. No. 1 Article 1, 2008.

[21] I.F. Cruz, A.O. Mendelzon, and P.T. Wood, "A graphical query language supporting recursion," Association for Computing Machinery Special Interest Group on Management of Data, ACM Press, 1987, pp. 323—330.

[22] M. Gemis, and J. Paredaens, "An object-oriented pattern matching language," 1st JSSST International Symposium on Object Technologies for Advanced Software. Springer-Verlag, 1993, pp. 339–355.

[23] R. Giugno, and D. Shasha, "GraphGrep: A fast and universal method for querying graphs," IEEE International Conference in Pattern recognition, 2002.

[24] InfoGrid: The Web Graph Database, http://infogrid.org/trac [retrieved: December, 2013].

[25] Apache Derby, http://db.apache.org/derby [retrieved: December, 2013].

[26] RamDisk Plus, http://www.raxco.com/home/ramdiskplus_workstation.aspx [retrieved: December, 2013].

[27] HyperSQL, http://hsqldb.org [retrieved: December, 2013].

[28] XING das professionelle Netzwerk, http://www.xing.com [retrieved: December, 2013].

Multi-Agent Distributed Data Mining by Ontologies

María del Pilar Angeles
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F.
pilarang@unam.mx

Jonathan Córdoba-Luna
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México
México, D.F.
jel_154@comunidad.unam.mx

***Abstract*—The present paper introduces a Multi-Agent Distributed Data Mining framework as an approach to performance and data security issues. It has been implemented by ontologies in order to incorporate semantic content to improve the intelligence and efficiency of Data Mining Agents. Each agent is only responsible for specific duties. Agents communicate and coordinate with each other to enhance data mining and keep privacy and confidentiality of data. The developed prototype shows a parallel, distributed data mining process, and a real-world use case, which integrates birth rate data registered during 2011-2012 in México by the official censuses.**

Keywords- distributed data mining; multi-agent system; inter-agent negotiation; ontologies; agent based distributed data mining

I. INTRODUCTION

The Process of Knowledge Discovery (KDD) is a set of processes focused on the discovery of knowledge within databases, while data mining is the application of a number of artificial intelligence, machine learning and statistics techniques to data. Data Mining is one of the most important processes within KDD. However, data mining is a computationally intensive process involving very large datasets, affecting the overall performance.

Distributed Data Mining (DDM) has emerged as an approach to performance and security issues because DDM avoids the transference across the network of very large volumes of data and the security issues occasioned from network transferences.

We have developed a Multi-agent Distributed Data-mining System also known as Multi-Agent Data Mining (MADM) to improve performance in [1].

According to Sumathi and Sivanandam in [2] data mining is related to the process of discovery of new and significant correlations, patterns and tendencies mined from very large data sources by using statistics, machine learning, artificial intelligence and data visualization techniques.

We consider data mining as the process of extraction of new and useful information from very large data sources by considering a number of multidisciplinary technics, such as statistics, artificial intelligence and data visualization aimed to make informed decisions that provide business advantage.

The discovered patterns must be meaningful enough to provide a competitive advantage, mainly in terms of business. However, in [3], Han proposed data mining as a complex data set analysis aimed to discover unsuspected data interrelations in order to summarize or classify data in different and understandable forms that should be useful to the data owner.

This approach is focused on improving the process of data mining; on reducing the exchange of messages on the sites that make up the DDM system; on keeping performance with respect to memory and CPU at sites containing limited resources; on showing that the developed prototype can be used to evaluate various data mining scenarios and for the data mining on a real-world use case.

In this paper, we present the definition of a number of ontologies in order to incorporate semantic content and more information to the messages exchanged between agents and thus by increasing interaction among agents they are able to make better decisions on the execution of clustering.

The present paper is organized as follows: The next section is focused on the process of data mining. The third section details cluster analysis by describing the K-Means and the agglomerative hierarchical algorithms. The forth section describes the performance problems related to data mining. Sections II, III and IV are aimed to describe the background of data mining and multi-agent systems.

Section V presents the proposed framework describing the multi-agents, the scope and limitations of the agents besides a set of criteria to assess the algorithms performance within a multi-agent based system architecture. Section VI is concerned to the implementation of the proposed framework, and the ontologies introduced.

Section VII shows the experimentation plan, which has considered four possible scenarios for the analyses of the experiment results in order to determine prototype performance.

Section VIII presents a case study of birth rate occurred and registered during 2011-2012 in Mexico. The last section concludes the main topics achieved and the future work to be done.

II. THE PROCESS OF DATA MINING

The present section is aimed to briefly describe the related work on data mining.

The process of data mining focuses on two main objectives: prediction and description. The main goals within a knowledge discovery project should be already determined and they will determine if descriptive or predictive models would be applied.

The availability of an expert or supervisor would determine the type of learning (supervised or unsupervised) that will apply during the data mining process. The predictive model learns under the control of a supervisor or expert (supervised learning) who determines the desired

answer from the data mining system [2], whereas the descriptive model executes clustering and association rules tasks to discover knowledge by unsupervised learning, in other words, with no external influence that establish any desired behaviour within the system [2].

The next task within data mining is the identification of methods and their corresponding algorithms for classification, clustering, regression analysis, or any other method that allows building a model that describes and distinguishes data within classes and concepts.

Classification is used mostly as a supervised learning method, whereas clustering is commonly used for unsupervised learning (some clustering models are for both). The goal of clustering is descriptive; that of classification is predictive [4].

III. CLUSTER ANALYSIS

As our proposal has been implemented with no external supervision, Section III is aimed to briefly explain only the implemented algorithms and metrics involved in our clustering analysis.

The term cluster analysis encompasses a number of different algorithms and methods for grouping objects of similar kind into respective categories. Such algorithms or methods are concerned with organizing observed data into meaningful structures. In other words, cluster analysis is an exploratory data analysis tool, which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Given the above, cluster analysis can be used to discover structures in data without providing an explanation or interpretation.

There are a number of classifications of clustering algorithms; this research takes a basic but practical classification that allows organizing the existing algorithms. Such algorithms are divided into two categories: Partition based algorithms and hierarchical algorithms.

A. Partition based clustering algorithms

Given a data set with n data objects to identify k data partitions, where each partition represents a cluster and $k \leq n$. There is a good partitioning if the objects within a cluster are close to each other (cohesion), or they actually are related to each other, and at the same time they are far from the objects that belong to another cluster. This section will explain the partition based clustering k-means algorithm [10].

The k-means algorithm represents each cluster by the mean value of the data objects in the cluster.

Given an initial set of k means (centroids) $m_1^{(1)},\ldots,m_k^{(1)}$, the algorithm proceeds by alternating between three steps:

1. Assignment step: Assign each observation to the cluster with the closest mean.
2. Update step: Calculate the new means to be the centroid of the observations in the cluster.
3. The algorithm is deemed to have converged when the assignments no longer change.

K-means is a classical partitioning technique of clustering that clusters the data set of n objects into k clusters with k known a priori.

Many clustering methods use distance measures to determine the similarity or dissimilarity between any pair of objects. It is useful to denote the distance between two g-dimensional instances x_i and x_j as: $d(x_i, x_j)$. A valid distance measure should be symmetric and obtains its minimum value (usually zero) in case of identical vectors. This section describes three distance measure for numeric attributes: Minkowski, Euclidean and Manhattan. The distance of order g between two data instances can be calculated using the Minkowski metric [5].

$$d(x_i, x_j) = (|x_{i1}-x_{j1}|^g+|x_{i2}-x_{j2}|^g+\ldots+|x_{ip}-x_{jp}|^g)^{1/g} \quad (1)$$

All distances obeying (1) are called Minkowsky distances. However, for g greater or equal to 1, these distances are also metrics. The Euclidean distance between two objects is achieved when g = 2, if g = 1 then the Manhattan distance is obtained.

B. Hierarchical clustering algorithms

These algorithms consist of joining two most similar data objects, merge them into a new super data object and repeats until all merged. There is a graphical data representation by a tree structure named dendrogram to illustrate the arrangement of the clusters produced by hierarchical clustering. There are two ways of creating the graphic, the agglomerative algorithm or divisive algorithm [5]. Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc.

The key operation of agglomerative hierarchical clustering algorithm is the computation of the proximity between two clusters. However, cluster proximity is typically defined with a particular type of cluster. The cluster proximity in this section will refer to the single link, complete link and group average respectively.

For the single link, the proximity of two clusters A, B is defined as the minimum of the distance (maximum of the similarity) between any two points x, y in the two different clusters. For the complete link, the proximity of two clusters A, B is defined as the maximum of the distance (minimum of the similarity) between any two points x,y in the two different clusters. For the group average, the proximity of two clusters Cx and Cy are of size Sx and Sy, respectively, is expressed as the average pairwise proximity among all pairs of points in the different clusters.

C. Clustering Evaluation

In most cases, a clustering algorithm is evaluated using internal, external and manual inspection: a) In the case of internal evaluation there are some measures like cohesion, separation, or the silhouette coefficient (addressing both, cohesion and separation); b) For external evaluation measures like accuracy, precision are utilized. In some cases, where evaluation based on class labels does not seem viable; c) careful (manual) inspection of clusters shows

them to be a somehow meaningful collection of apparently somehow related objects [6].

There are a number of important issues for cluster validation, such as the cluster tendency of a set of data, the correct number of clusters, whereas the cluster fit the data without reference to external information or not, and determining which clustering is the best [7]. The first three issues do not need any external information.

The evaluation measures are classified into unsupervised, supervised and relative. We have implemented the unsupervised evaluation.

Unsupervised validation: In the case of cluster cohesion is concerned to how closely relate the objects in a cluster are. In the case of cluster separation is aimed to determine how distinct a cluster is from other clusters, these internal indices use only information from the data set [7].

Cluster Cohesion: Measures how closely related are objects in a cluster. Then, cluster cohesion can be defined as the sum of the proximities to the cluster centroid or medoid.

Cluster Separation: Measures how distinct or well-separated a cluster is from other clusters. Therefore, cluster separation is measured by the sum of the weights of the links from points in one cluster to points in the other cluster.

Given a similarity matrix for a data set and the cluster labels from a cluster analysis, it is possible to compare this similarity matrix against an ideal similarity matrix on the basis of cluster labels. An ideal cluster is one whose points have a similarity of 1 to all points in the cluster and a similarity of 0 to all points in other clusters.

In the case of unsupervised evaluation of hierarchical based clustering algorithms, we discuss the cophenetic correlation.

In the agglomerative hierarchical clustering process, the smallest distance between two clusters is assigned, and then all points in one cluster will have the same value as a cophenetic distance with respect to the points in other cluster. In a cophenetic distance matrix, the entries are the cophenetic distances between each pair of objects.

If any of single link clustering, complete link or group average is applied, the cophenetic distances for each point can be expressed in cophenetic distance matrix. Thus, the cophenetic correlation coefficient is the correlation between the entries of this matrix and the original dissimilarity matrix and is a standard measure of how well a hierarchical clustering fits the data. As we have briefly described, data mining requires the execution of complex algorithms, bringing some performance issues as a consequence. These issues will be mentioned in the following section.

IV. PERFORMANCE PROBLEMS ON DATA MINING

As we have mentioned in previous sections, many methods exist for data analysis and interpretation. However, these methods were often not designed for the terabyte sizes of large data sets data mining is dealing with today. There are significant issues of scalability and efficiency of the data mining methods when processing considerably large data. Algorithms with exponential and even medium-order polynomial complexity cannot be of practical use for data mining. Linear algorithms are usually the norm. In same theme, sampling can be used for mining instead of the whole dataset. However, concerns such as completeness and choice of samples may arise. Other topics in the issue of performance are incremental updating, and parallel programming. There is no doubt that parallelism can help solve the size problem if the dataset can be subdivided and the results can be merged later. Incremental updating is important for merging results from parallel mining, or updating data mining results when new data becomes available without having to re-analyse the complete dataset [8].

In the 1990s, Bailey proposed in [9] a multi-agent clustering system to achieve the integration and knowledge discovered from different sites with a minimum amount of network communication and maximum amount of local computation by a distributed clustering system where data and results can be moved between agents. There was proposed a distributed density based clustering algorithm the Peer to Peer model in [10]

These previous approaches were aimed to improve security by a distributed data mining. However, there were no measurements of general performances by considering distributed agents against centralized clustering techniques within a data warehouse.

In order to improve performance and to implement parallelism we have proposed the use of multi-agent system within a distributed data mining system. We are considering the following database oriented constraints: a limited acceptable response time, maximum resource optimization, maximum adjust to available memory, minimum I/O costs.

V. MULTI AGENT SYSTEM FOR DISTRIBUTED DATA MINING FRAMEWORK

This section is focused on the description of the Framework we have proposed for the Multi-agent Distributed Data Mining System.

A. Introduction

Multi-agent system has revealed opportunities to improve distributed data mining in a number of ways in [11]. However, a single data mining technique has not been proven appropriate for every domain and data set [11].

An agent is a computer system that is capable of autonomous action on behalf of its user or owner. An agent is capable to figure out what it is required to be done, rather than just been told what to do [12].

An intelligent agent must be reactive, pro-active, and social. A reactive agent maintains an ongoing interaction with its environment, and responds in time to changes that occur in it. A proactive agent attempts to achieve goals, not only driven by events, but also taking the initiative. However, at the same time a social agent takes into account the environment, in other words, some goals can only be achieved by interacting with others. The social ability in agents is the ability to interact with other agents (and possibly humans) via cooperation, coordination, and negotiation. Agents have the ability to communicate, to cooperate by working together as a time to achieve a shared goal. Agents have the ability to coordinate different

activities. Agents will negotiate to reach agreements taking into consideration the environment in order to react, to negotiate, to coordinate, etc. The environments are divided in accessible, inaccessible, deterministic, non-deterministic, episodic, static and dynamic.

A multi-agent system is one that consists of a number of agents, which interact with one another.

We propose a mining task that involves a number of agents and data sources. Agents are configured to choose an algorithm and deal with given data sets. Furthermore, performance can be improved because mining tasks can be executed in parallel.

The present research proposes a framework for a Multi-agent Distributed Data mining, based on models presented in [13] and [14], besides such framework has been implemented and extended by additional agents such as performance, validating and coordinating agents in order to address performance and security issues within the disparate information systems that conform the distributed data mining system.

Our approach also proposes the use of ontologies to improve inter-agents communication by sharing the same language, vocabulary and protocols. Therefore, intelligent agents for distributed data mining would be able to improve the data mining process by a more informed and better decision making. For instance, intelligent agents would be able to handle the access to the underlying data sources according to specific security constraints. Pro-active agents would decrease human intervention during data mining process; they may adaptively select data sources according to given criteria, such as the type, quality or expected amount of data. Intelligent agents allow performing mining tasks locally to each of data sites and may evaluate the best strategy between working remotely or migrating data sources [14].

B. Agents

The proposed framework is composed by a number of agents, which are described as follows.

a) A user agent is responsible for the interaction between end-users and the coordinating agents in order to accomplish the assigned tasks.

b) Coordinating agent is focused on the correct message transmission among the agents within the network. It takes the user requirements and sends them to the corresponding agent.

c) Coordinating Algorithm Agent is focused on the interaction between clustering agents. This agent receives the processed information from the clustering agents and executes the algorithm globally in order to guarantee a better clustering quality.

d) Clustering agent is concerned with a clustering algorithm. Once the clustering agents have done their task, they send local processed information to the algorithm coordinator agent. The clustering algorithms are the most commonly used and keep the same structure utilized within a centralized approach but they can be sent to other sites where is required to perform clustering avoiding data transference in order to enhance performance and enforce security.

e) Data agent is in charge of a data source; it interacts and allows data access. There is one data agent per data source.

f) Validation agent is responsible for the quality assessment of the clustering results. There is one validation agent per a measuring technique of a given cluster configuration. These agents consider either cluster cohesion or cluster separation. In the case of the hierarchical clustering, the cophenetic distance is utilized to measure the proximity within the hierarchical agglomerative clustering algorithm. This distance helps to determine the precision. Therefore, is required to compute the similarity matrix and the cophenetic matrix. The cophenetic distance can be seen as a correlation between the distance matrix and the cophenetic matrix. If the computed value is close to 100%, the quality of clustering is enough.

g) Performance agent is focused on the measurement of operating system resources in order to obtain the overall performance of the processing algorithms in terms of data transmission, data access and data process.

C. Measurement and Assessment Performance

In order to assess performance during data-mining, we have considered the following metrics:

a) Memory used: physical memory consumed by the algorithm when it has been executed. The resulting value is given in megabytes (MB).

b) Elapsed Processing Time: the amount of time the algorithm took to process. The resulting value is given in nanoseconds (ns).

c) Amount of data transmitted: A quantity in MB to determine the total size of all data processed and transferred.

d) *PC-LAN Broadband*: Amount of information that can be sent over a network connection in a given period of time. The bandwidth is usually given in bits per second (bps), kilobits per second (kbps) or megabits per second (mps).

e) Elapsed response time: Time interval from which the request is made by the user until the result set is presented to user.

f) Transmission-time: time of the node-to-node data transfer.

g) Total Response Time: The total result of the processing time + transmission time + response time.

h) Physical reads: total number of data blocks read from disk

i) Logical reads: total number of data blocks read from the main memory (RAM/cache).

All these measures are stored within a table as a log from where the data agent can access and inform the performance agent. Therefore, when a user request is submitted, it will be evaluated according to the historical information stored in the log, and an execution strategy will be developed.

If the amount of data to be processed is small, the performance agent will establish a "low status", thus the creation of a single clustering agent to perform clustering analysis would be enough.

If the amount of data is considerably high, the performance agent establishes a "medium status", in order to create two agents to process the data and obtain the clustering analysis.

If the amount of data is very large, the performance agent establishes a "high status", in order to create three clustering agents for clustering analysis.

The status is sent to the coordinating agent, which is responsible for building the agents requested.

In order to improve the clustering results and the performance of data mining across the distributed system, there has been implemented negotiation among agents by a communication protocol. For instance, considering the amount of data to analyse, there is a negotiation of what clustering method is the best by asking each clustering agent if it is able to perform the task according to the resources of the site where that agent resides.

The framework proposes a performance agent which, according to the status established from negotiation and statistics, it is able to determine the strategy to implement the algorithms through clustering agents running on parallel.

Fig. 1 shows the Multi-Agent System for Distributed Data Mining Framework.

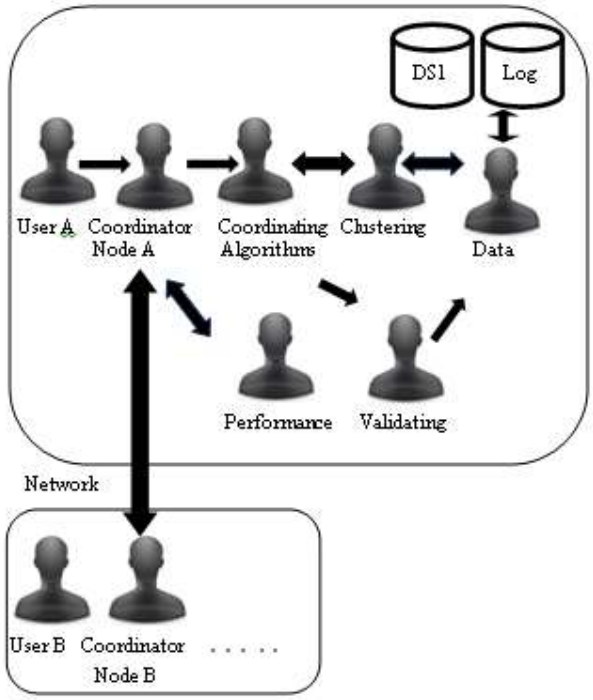


Figure 1. Multi-Agent System for Distributed Data Mining

VI. IMPLEMENTED FRAMEWORK

The present work proposes the implementation of the Multi-Agent System for Distributed Data Mining framework described in previous section. We have developed a web platform through Agent-Oriented Programming paradigm (AOP).

In order to allow inter-agents communication, agents must share the same language, vocabulary and protocols. In order to achieve so, we have followed the recommendations of the standard Foundation for Intelligent, Physical Agents (FIPA). However, one must define specific ontologies, with its own vocabulary and semantics of the content of the messages exchanged by the agents. We have developed our proposed framework with Java Agent DEvelopment (JADE) [15], which integrates a library called "jade.gateway" for the agent programming within a web interface. The following section briefly describes the FIPA Communication Acts and Semantic Language.

A. FIPA Communication Acts and SemanticLanguage

JADE is compliant to the FIPA[16]. FIPA specifications represent the most important standardization activity conducted in the field of agent technology. JADE is composed by a native Agent Communication Language (ACL), which incorporates an Agent Manager System (AMS) and a Directory Facilitator (DF).

JADE provides three different ways to carry out communication between agents:

1) The use of strings to represent the content of the messages. This alternative is convenient when message contents are atomic data. However is not useful in the case of abstract concepts or structured data objects because string parsing would be required to access each component.

2) The use of Java serializable objects, which directly transmit message contents. This option is suitable in case of local applications where all agents are implemented in Java. However, messages are not human understandable.

3) The definition of objects to be transferred as an extension of the predefined JADE classes in order to encode or decode the messages into a FIPA standard format. This alternative allows JADE agents to interoperate with other agent systems. This feature has been implemented in our prototype.

The Agent Communication Language may be modified according to system requirements. Message Transport Service (MTS) is a service provided to transport FIPA-ACL messages between agents in any given agent platform and between agents on different agent platforms. The Agent Management System is responsible for managing the operation of an agent platform, such as the creation, deletion, status, overseeing and migration of agents. The Directory Facilitator provides yellow pages services to other agents, maintaining a list of agents and providing the most current information about agents in its directory to all authorized agents.

In order to implement negotiation among agents, we have utilized a number of communicative acts and protocols for effective inter-agent communication:

OneShotBehaviour: This type of behaviour is executed only once and with no interruption; CyclicBehaviour: Represents a behaviour that should be executed a number of times; CompositeBehaviour: Behaviour based on the composition of other behaviours or sub-behaviours, the implementation of the framework proposed contains the following CompositeBehaviour subclasses; SequentialBehaviour: executes a series of sub-behaviours sequentially, and is considered finished when all its sub-behaviours have been completed. ParallelBehaviour: executes a series of behaviours concurrently and ends when a certain condition is met upon completion of the sub-behaviours:

The following communication protocols have been implemented:

FIPA-Request: Allows an agent to request another agent to perform an action. The messages exchanged are:

"Request" followed by the request, "Agree", if the request is accepted, "Refuse" in case the request is rejected. "Failure", if an error occurred in the process, "Inform", to communicate the results.

FIPA-Query: Allows an agent to request another agent an object by a "Query-ref()" message or a comparison value by an if() message, depending on what type of request it will be a query-if (test of truth). The messages exchanged are: "Agree", Refuse", "Failure" and "Inform".

The class ContractNet implements protocol behaviour where an initiator sends a proposal to several responders and select the best proposal. The messages exchanged are Call For Proposal (CFP) in order to specify the action to perform. Therefore, the responders may send a "Refuse" to deny the request, a "Not-Understood" if there was a failure in communication, or "Propose" to make a proposal to the originator. The initiator evaluates the proposals received and sends "Reject-Proposal" or "Accept-Proposal. Responders whose proposal was accepted send a "Failure" if something went wrong, an "Inform-Done" if the action was successful or an "Inform-Result" with the results of the action if appropriate.

B. Ontologies for inter-agent communication

The development of Multi-agent Systems is not an easy task; there are a number of issues related to these implementations, such as high network traffic derived from communication between agents, problems related to interoperability of systems and platforms and semantic problems.

The inherent complexity of the applications developed in the context of Multi-Agent Systems requires the use of ontologies.

In order to allow agents to communicate each other, they must share the same language, vocabulary and protocols. By following the recommendations of the standard FIPA, JADE already provides a certain degree of overlap when using FIPA communicative acts and content language SL (Semantic Language), which determines how messages are exchanged by the agents. However, one must define specific ontologies, with its own vocabulary and semantics of the content of the messages exchanged by the agents.

The term ontology is concerned to the description of concepts and the relationships between them. The ontologies form part of the knowledge of an agent or a society of agents.

Ontology is defined within JADE in order to improve the communication among agents. An agent who wants to communicate with other agents within a given application domain, should have a common ontology to those agents that define the terms to be used. This allows agents to make more informed decisions.

By using ontologies we have incorporated semantic content and data to the messages exchanged between agents. However, as ontologies are defined based on Java objects, semantic is required to be encapsulated or encoded within ACL messages.

C. Conversion support for ontologies.

Jade incorporates in the *jade.content* package, support (codecs) for two content languages:

The language SL is human readable and encoded as string expressions, and the LEAP language, which is not readable by humans and is byte-encoded.

Ontology is an instance of the class *jade.content.onto.Ontology* where schemas are defined. Schemas are sets of elements that define the structure of the predicates, the agent actions and concepts relevant to the problem domain. We explain these concepts as follows:

- Predicate: expressions on the state of world. Typical applications INFORM messages and QUERY-IF, not REQUEST.
- Agents Actions: expressions that indicate the actions some agents can perform. Typically used in REQUEST type messages.
- Concepts: expressions representing objects, representing a structure with several attributes. No messages appear isolated but included in other items.
- Other elements: primitive (atomic elements as numbers or strings), aggregations (sets, lists of other terms), expressions (identified entities for which a predicate is true), variables.

We have identified and defined a number of Concepts, Agents Actions and predicates in order to establish a formal vocabulary for inter-agent communication.

D. Implementation of Ontology within the Distributed Data Mining Based on Multi-Agent Systems:

As we have mentioned before, our prototype has implemented the following agents: User Agent, Coordinator Agent, Data Agent, Manager Agent Algorithms, Performance Agent, Clustering Agent and Validation Agent.

We have defined several packages in order to allow inter-agents communication. Each package is composed by concepts, agent actions and predicates. Such packages are mentioned as follows:

a) Algorithm Ontology

This package contains the ontology to communicate the User Agent with the Agent algorithm.

b) Data Ontology

This package contains the ontology to communicate the Coordinating Agent or the Coordinating Algorithm Agent with the Data Agent.

c) Strategy Ontology

This package contains the ontology to communicate the Coordinating Agent or the Coordinating Algorithm Agent with the Performance Agent and get a status.

d) Activity Ontology - Part A

This package contains the ontology to communicate the Coordinating Agent with the Coordinating Algorithm Agent.

e) Measures Ontology

This package contains the ontology to communicate the Performance Agent with the Data Agent.

f) Validation Ontology

This package contains the ontology to communicate the Coordinating Algorithm Agent with the Validation Agent.

g) Clustering Ontology

This package contains the ontology to communicate the Coordinating Algorithm with the Clustering Agent.

h) DataSource Ontology

This package contains the ontology to communicate the Coordinating Algorithm or Coordinating Algorithm Agent with the Data Agent. The following section describes the Web application architecture of the prototype implemented for the Multi-agent Distributed Data mining system.

E. Web Application architecture:

The Multi-agent System for Distributed Data mining Framework has been developed as a web application in order to be available for the all users within the network. The application is composed by a web interface, data repositories, clustering repository and the system engine, which are presented in Fig. 2.

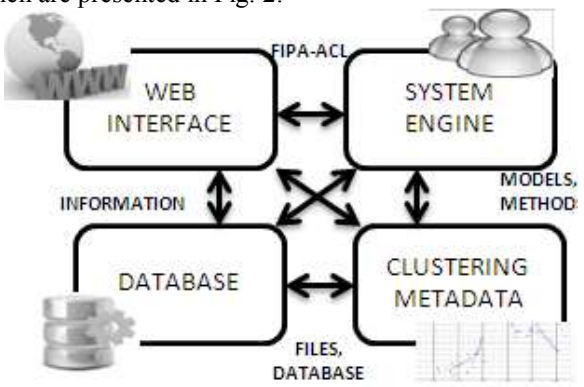


Figure 2. Web Application Architecture

a) The Web interface allows users to interact with the Multi-Agent System through a web browser by sending request of data mining tasks and receiving the corresponding results.

b) Data repositories, which consist of file folders or PostgreSQL databases.

c) Clustering Repository with all the clustering and validation algorithms.

d) The System engine for the involved agent management, data pre-processing, connection to the Database Management Systems (DBMS), and sites communication languages.

The web interface calls the user agent to allow users the specification of the node and the data source from which the clustering is required.

User agent asks the data agent to connect to the distributed database system and to retrieve information from a specific database table or file within a remote or local site.

Once the node has been specified, the database and table the data mining system requires the specification of the clustering algorithm, the K number of clusters and the metric.

Fig. 3 shows partial results of the execution of the K-means algorithm with 5 clusters and the metric Euclidean distance.

PC: PC 1 DB: mining
Relation: weights
Results of the algorithm: K-Means
Metric Used: Euclidean Distance

Patterns	Cluster
V = [1.0, 95.2]	1
V = [2.0, 100.2]	2
V = [3.0, 70.2]	3
V = [4.0, 75.7]	3
V = [5.0, 90.3]	1
V = [6.0, 84.9]	2
V = [7.0, 32.3]	2
V = [8.0, 56.7]	1
V = [9.0, 85.4]	1
V = [10.0, 40.2]	3

Figure 3. K-means with 5 clusters and Euclidian distance

VII. EXPERIMENTS AND RESULTS

In order to assess the framework proposed in Section V, we have carried out a set of experiments according to the following possible scenarios:

a) Centralized Data Scenario: A typical data mining system, composed by a centralized data mining process with no multi-agents.

b) Multi-agent Centralized Data Scenario: A Multi-agent centralized data mining system.

c) Distributed Scenario: A Distributed data mining system with no multi-agents.

d) Multi-agent distributed data mining Scenario: A Distributed data mining system with multi-agents.

The identified independent variables are: a) clustering methods; b) metrics; c) number of clusters; d) data sources

The identified dependent variables are: a) data access time; b) data transmission time; and c) processing time.

For each scenario a set of 9 data sources have been processed, the corresponding results are presented as follows:

a) Centralized data scenario

Table I presents the results obtained from processing 9 data sources by the k-means algorithm, considering no agents, 10 clusters and a transfer rate of 500 kb/s. For instance, the process of mining a table called agency with 35000 rows takes 7.83E+09 nanoseconds, and 7.11 Mb of memory used.

TABLE I. CENTRALIZED, K-MEANS, 10 CLUSTERS SCENARIO

Table name	Rows	Data Transfer (Mb)	Data Transfer Time (ns)	Memory Used (Mb)	Processing Time (ns)
agency	35000	0.200272	3.13E+08	7.11	7.83E+09
school	500	0.003893	6.08E+07	1.22	3.08E+08
supermarket	150	0.001001	1.56E+07	1.10	3.06E+08
weights	70	0.000476	7.44E+06	0.76	2.77E+08
substance	800	0.003338	5.22E+07	1.31	4.36E+08
articles	500	0.002538	3.97E+07	1.22	3.56E+08
survey	300	0.005728	8.95E+07	1.51	3.13E+08
population	300	0.002251	3.52E+07	1.15	2.87E+08
school_age	1200	0.008817	1.38E+08	1.45	5.44E+08

Table II presents the results obtained from processing 8 data sources by the hierarchical algorithm, considering no agents and 10 clusters.

TABLE II. CENTRALIZED, HIERARCHICAL, 10 CLUSTERS SINGLE LINK SCENARIO

TableName	Rows	Processing Time
school	500	7.15E+08
supermarket	150	3.89E+08
weights	70	2.33E+08
substance	800	1.69E+09
articles	500	6.80E+08
survey	300	4.33E+08
population	300	4.31E+08
school_age	1200	4.28E+09

b) Multiagent centralized data

Table III presents the results obtained from processing 9 data sources by the k-means algorithm, considering multi-agents and 10 clusters. For instance, the process of mining a table called agency with 35000 rows takes 7790887000 nanoseconds.

TABLE III. MULTI-AGENT, CENTRALIZED, K-MEANS, 10 CLUSTERS SCENARIO

TableName	Rows	Processing Time
agency	35000	7.79E+09
school	500	2.74E+08
supermarket	150	2.71E+08
weights	70	2.43E+08
substance	800	4.02E+08
articles	500	3.21E+08
survey	300	2.79E+08
population	300	2.53E+08
school_age	1200	5.10E+08

Table IV presents the results obtained from processing 8 data sources by the hierarchical algorithm, considering no agents and 10 clusters.

TABLE IV. MULTI-AGENT, CENTRALIZED, HIERARCHICAL, SINGLE LINK, 10 CLUSTERS SCENARIO

TableName	Rows	Processing Time
school	500	6.81E+08
supermarket	150	3.55E+08
weights	70	1.99E+08
substance	800	1.66E+09
articles	500	6.46E+08
survey	300	3.99E+08
population	300	3.97E+08
school_age	1200	4.25E+09

c) Distribuited data scenario

Table V presents the results obtained from processing the Agency table distributed on two partitions stored on node A and node B. The Agency table was processed by the k-means algorithm, with no consideration of agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 775756400 nanoseconds agency.

TABLE V. DISTRIBUTED AGENCY TABLE ON TWO PARTITIONS, NO AGENTS SCENARIO

Data rows Node A	Data rows Node B	Total Processing Time
18000	18000	7.76E+08

d) Multi-agent distributed data mining scenario

Table VI presents the results obtained from processing the Agency table distributed on two partitions stored on Node1 and Node2. The Agency table was processed by the k-means algorithm, with multi-agents. For instance, the process of mining 36000 rows by the k-means algorithm takes 748213000 nanoseconds agency.

TABLE VI. MULTI-AGENT, DISTRIBUTED AGENCY TABLE, 2 PARTITIONS

Data rows Node1	Data rows Node 2	Total Time Processing
18000	18000	7.48E+08

Table VII presents the results obtained from processing a set of 9 data sources, three agents, three partitions within a

distributed environment, and clustering algorithm k-means. The memory used for each agent is also presented.

TABLE VII. MULTI-AGENT, DISTRIBUTED, K-MEANS

Table name	Number of Rows	Memory Used Agent 1	Memory Used Agent 2	Memory Used Agent 3	Memory Used Total
agency	35000	2.33	2.33	2.33	6.99
school	500	0.36	0.36	0.36	1.08
supermarket	150	0.33	0.33	0.33	0.99
weights	70	0.21	0.21	0.21	0.63
substance	800	0.40	0.40	0.40	1.20
articles	500	0.36	0.36	0.36	1.08
survey	300	0.34	0.34	0.34	1.02
population	300	0.34	0.34	0.34	1.02
School_age	1200	0.44	0.44	0.44	1.32

e) Analysis of Results

According to the identified four scenarios, and in order to justify the use of multi-agents for the performance improvement, we present in this section a comparison of CPU processing time and memory utilization in terms of the results we have obtained. Fig. 4 shows a CPU processing time advantage in the use of multi-agent system against no agents system for clustering 8 datasources with the K-means algorithm. Processing the data partitions with multi-agents and merging the results allows faster data processing. If the amount of data is significantly large, data can be shared among n agents, reducing response time. However, a disadvantage could be that by sharing data between n agents the quality of the clusters may decrease.

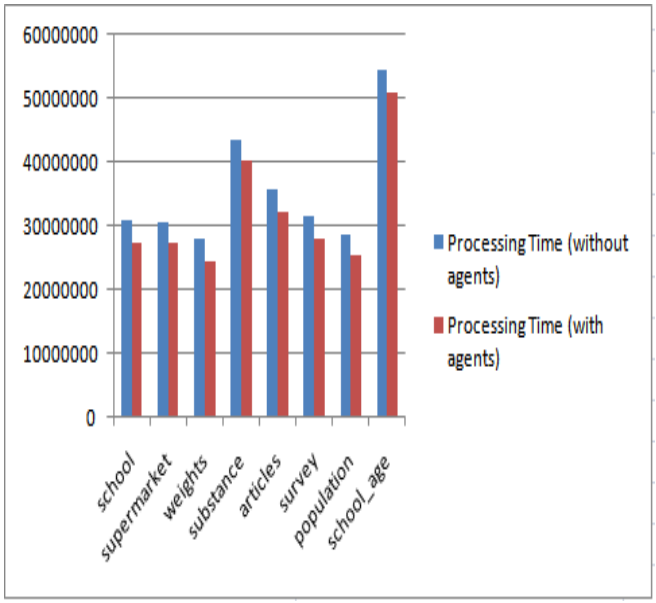


Figure 4. Centralized no agents vs. multi-agents with k-means algorithm

Fig. 5 compares the four scenarios identified in terms of CPU processing. The comparison shows the advantage obtained from clustering the distributed Agency table with 35000 rows on two partitions versus centralized data and furthermore, the advantage of using multi-agents system against no agents system in terms of cpu time for the same data source

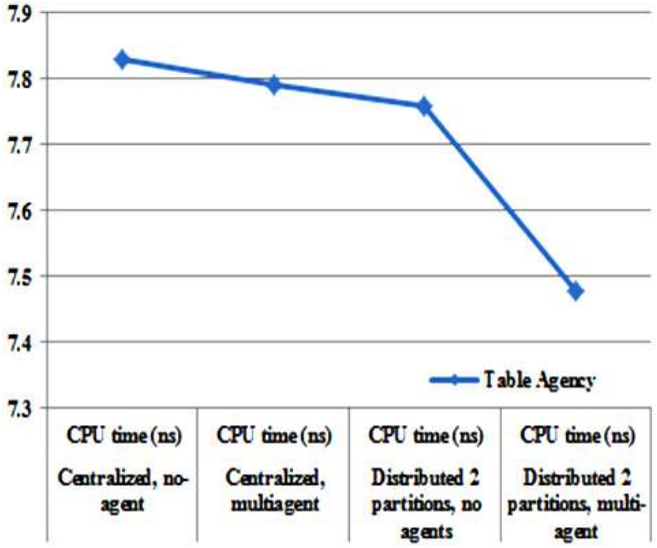


Figure 5. CPU processing time, K-means, four case scenarios.

Fig. 6 compares the four scenarios identified in terms of memory utilization. The comparison shows a slight advantage obtained from clustering the 9 data sources on three partitions versus centralized data and furthermore, the advantage of using multi-agents system against no agents system in terms of memory for the same data sources. Therefore, we can conclude that the amount of memory used in multi-agent, distributed environment was less than the memory required for the no-agent, centralized environment in all cases.

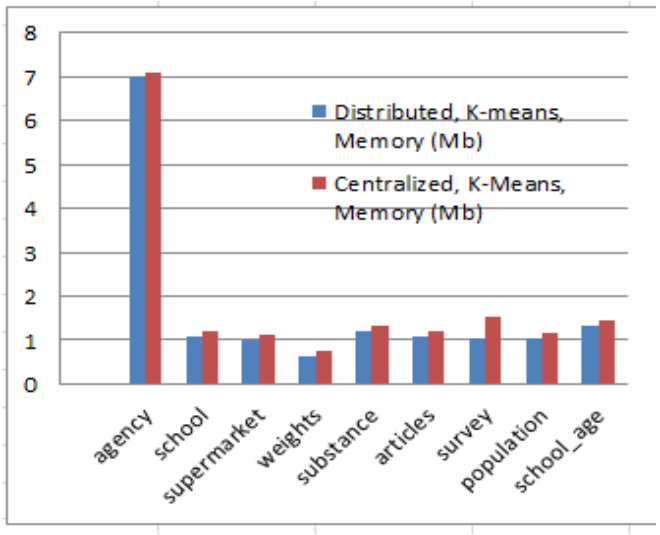


Figure 6. Distributed vs. centralized clustering in terms of memory .

If we consider that the total amount of memory utilized in three sites is less than the total amount required in only one site, we can conclude that is possible to achieve a balanced workload and a better utilization of resources,

because they are distributed among several sites and be executed in parallel in order to obtain better response time.

We can conclude that agents reduce CPU time processing, memory utilization and response time by the utilization of multi-agents and distributed data. Furthermore, negotiation and parallelization of agents is recommended. Even the reduction has not been very significant, the proposal pointed out that distributed data mining algorithms may offer a better solution since they are designed to work in a distributed environment by paying careful attention to the computing and the communication resources.

We have achieved data privacy within a distributed multi-agent scenario, where data are processed locally and the result has been wrapped by another agent, allowing a significant data processing optimization under clustering algorithms.

There is a trade-off between the clustering accuracy and performance due to the cost of the computation. On the one hand, if the interest is accurate clustering, is better to transfer all data to a single node and execute the clustering with the whole information. On the other hand, if the interest is performance in terms of computation and communication costs, is better to execute clustering data locally obtaining local results, and combine the local results at the requesting node to obtain the final result. We assume that in general, this is the less expensive while the former approach is more accurate, but more expensive.

Once the Multi-Agent Distributed Data Mining System has been tested, we have carried out a data mining process as a case study of birth rate registered during 2011-2012 in México.

VIII. A CASE STUDY OF BIRTHRATE

A. Census Database Description and Preprocesing

The present section shows a specialized data mining process, which integrates birth rate data registered during 2011-2012 in México by the official censuses National System of Health Information "*Sistema Nacional de Información en Salud" (SINAIS)*. This birth rate database is comprised of a total of 64 variables; such variables were transformed into numerical values. Some numerical variables were eliminated, leaving a total of 55 variables.

The data mining was processed through the K-means algorithm and 10 clusters.

B. Birth Rate Analysis

The Multi agent distributed Data mining system is aimed to the generation of patterns of interest based on the clustering of districts with low birth rates for different causes of death in México. The following section is focused on the analysis of the clustering obtained. Fig. 7 shows the clustering results by K-means algorithm.

According to the results of the clustering process, we can conclude the following:

In the first cluster, two infants were born in the state of Aguascalientes and in the same locality. So, in this case, the classification was made according to the entity of birth.

In reference to the second cluster, most people are married or living common-law, most of this population had 1 or 2 children born dead, but in the current parity new-borns born alive. In most cases, the mothers received prenatal care even though most of them are not entitled to any health unit service. Infants received most of their vaccinations and vitamins.

With respect to the third cluster, continue to dominate the case of mothers who are married or cohabiting, the special feature of this cluster is that the new-born populations were mainly male, and they were registered on the first day of the month, in 2011.

The forth cluster is related to mothers who received prenatal care in the second trimester of pregnancy and were entitled to the National Health Common Service. A particular feature of this cluster is that most mothers are working in education, but currently they are not working.

In the fifth cluster, there is the case of mothers who had 1 or 2 children born dead before, but in the current delivery, the child survived. The population has been entitled to the National Health Common Service or to the Mexican Institute of Social Security. However, the infant was not provided of any kind of vaccine or vitamin, in most cases. Most births were attended by midwives.

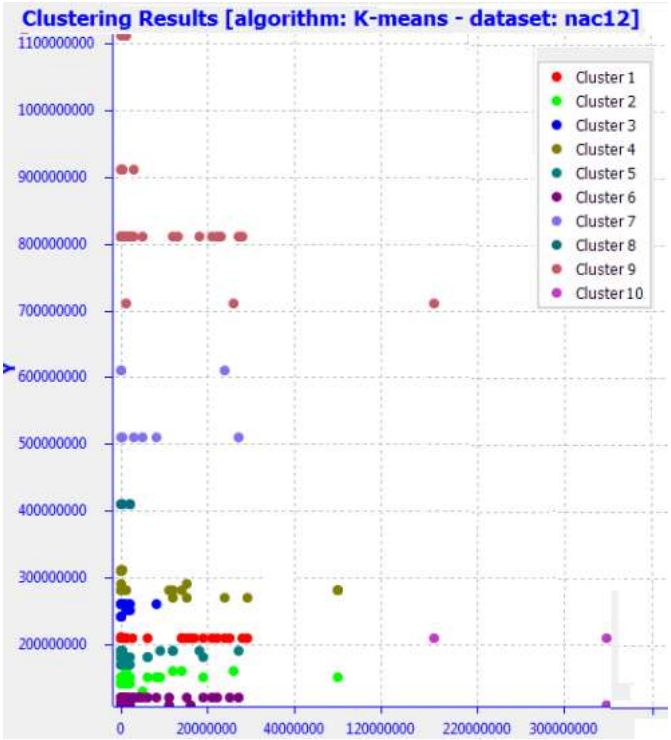


Figure 7. Birth rate data clustering by K-means with K=10.

The sixth cluster shows that in most cases mothers were housewives; in such cases, the infants were not given any kind of extra treatment, vitamin or vaccination.

The seventh cluster shows seven mothers living in the state of Aguascalientes that were attended by officials from the Ministry of Health. In this group, those women are

housewives whose infants did not receive any extra attention or necessary vaccinations.

Cluster 8 presents the case of mothers who have had 1-3 pregnancies where there has been a baby born dead, these mothers still being the case of housewives. But in this case they were grouped according to the attention they received from authorities of the Ministry of Health or a paediatrician.

Cluster 9 presents births that were certified in February. In most cases, the births were attended by a paediatrician, who had supplied vitamins and vaccines to the new-borns.

In the case of cluster 10, it shows the case of mothers whose status is single, married or cohabiting entitled or not to any Health Service. A particular feature of this group is that the majority of births took place in November 2011.

IX. Conclusion and future Work

Nowadays, organizations that operate at global level from geographically distributed data sources require distributed data mining for a cohesive and integrated knowledge. Such organizations are characterized by end users localized geographically separated from the data sources. The MDD is a relatively new research field, so a considerable number of research problems lie, relatively unaddressed.

Nowadays, k-means and agglomerative hierarchical clustering algorithms with their corresponding metrics such as Euclidean distance, Minkowski distance, Manhattan distance and single link are utilized. However, the present implementation could be improved by incorporating new algorithms.

The process of clustering can lose precision when data is partitioned and processed locally; the coordinating algorithm agent merges only the results into a single cluster in the case of hierarchical clustering algorithm. However, there is a better performance and cutbacks in memory space used.

We have proposed a Multi-Agent Distributed Data Mining System in order to improve data mining performance and data security considering inter-agent negotiation and metadata. This has allowed better decision regarding how many agents and where they are required by considering further information stored on metadata.

According to the experiments results, we can conclude that there is a better performance in terms of response time, memory utilization and processing distribution comparing with no agents and centralized environments.

We have incorporated semantic content and important data within the messages exchanged between the agents in order to improve inter-agents communication, better negotiation and, finally, an improvement on quality clustering.

Regarding the information stored within the log, the present implementation utilizes tables containing numerical data.

As part of future work, we have identified the following new research directions:

- The improvement of strategies for processing distributed clustering tasks. These strategies involve aspects of information organization, resource management and data analysis
- The development of agents in order to execute data pre-processing tasks, such as data cleaning, data integration, selection and data transformation
- The development of agents for the execution of further clustering tasks, such as density-based clustering and grid
- The development of agents for concurrency and distribution control, such as mobile agents
- The creation of further agents in order to transform data into numerical ratings

References

[1] P. Angeles, F.J. Garcia-Ugalde, and J. Cordoba-Luna, “Enhancing distributed data mining performance by mullti-agent systems,” Proc. The Fifth International Conference on Advances in Databases, Knowledge, and Data Applications, (DBKDA 2013), IARIA, 2013, pp. 174-181.

[2] S. Sumathi and S.N. Sivavavdam, ”Introduction to data mining and its applications,” Studies in Computational Intelligence, Springer Verlag, 2006, p. 828.

[3] J. Han, M. Kamber, and J. Pei, “Data mining: concepts and Techniques,” 3rd ed., Elsevier, p.744, 2011.

[4] M.P. Veyssieres and R.E. Plant, “Identification of vegetation state and transition domains in California’s hardwood rangelands,” University of California, p. 101, 1998.

[5] L. Gueguen and G.K. Ouzounis, “Hierarchical data representation structures for interactive image information mining,” International Journal of Image and Data Fusion, Special Issue: Image Information Mining for EO Applications, vol. 3, no. 3, 2012, pp. 221-241,doi:10.1080/19479832.2012.697924.

[6] H.P. Kriegel, E. Schubert, and A. Zimek, “Evaluation of multiple clustering solutions,” Universität München, Germany, 2013.

[7] P. Tan, M. Steinbach, and V. Kumar, “Introduction to data mining,” Addison-Wesley, Companion Book Site, 2006.

[8] O. Zaine, “Principles of knowledge discovery in databases, Chapter 1: Introduction to data mining,” University of Alberta, 2013.

[9] S. Bailey, R. Grossman, H. Sivakumar, and A. Turinsky, “Papyrus: a system for data mining over local and wide area clusters and super-clusters, IEEE Supercomputing, 1999.

[10] M. Klusch, S. Lodi, and G. Moro, “Agent-based distributed data mining: The KDEC Scheme,” Proc. Springer Lecture Notes in Computer Science, vol. 2586, 2003, pp. 104–122.

[11] M. Wooldridge, “An introduction to multiAgent systems”, 2nd ed., John Wiley & Sons, ISBN-10: 0470519460.

[12] S. R. Vuda, “Multi agent-based distributed data mining, an overview,” International Journal of Reviews in Computing, pp. 83-92, ISSN: 2076-3328, E-ISSN: 2076-3336.

[13] S. Chaimontree, K. Atkinson, and F. Coenen, “A multi-agent approach to clustering: Harnessing The Power of Agents,” Springer-Verlag, 2012, pp. 16-29.

[14] N.P. Trilok, P. Niranjan, and K.S.Pravat, “Improving performance of distributed data mining (DDM) with multi-agent system,” International Journal of Computer Science, vol. 9, no. 2& 3, 2012, pp. 74-82, ISSN:1694-0814.

[15] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, “JADE: a java agent development framework,” Multi-agent Programming: Languages, Platforms, and Applications, Springer-Verlag, 2005, p. 295.

[16] FIPA: Communicative Act Library Specification. Tech. Rep. XC00037H, Foundation for Intelligent Physical Agents, 2013.

Multi-Version Databases on Flash: Append Storage and Access Paths

Robert Gottstein
Databases and Distributed Systems Group
TU-Darmstadt, Germany
gottstein@dvs.tu-darmstadt.de

Ilia Petrov
Data Management Lab
Reutlingen University, Germany
ilia.petrov@reutlingen-university.de

Alejandro Buchmann
Databases and Distributed Systems Group
TU-Darmstadt, Germany
buchmann@dvs.tu-darmstadt.de

***Abstract*—New storage technologies, such as Flash and Non-Volatile Memories, with fundamentally different properties are appearing. Leveraging their performance and endurance requires a redesign of existing architecture and algorithms in modern high performance databases. Multi-Version Concurrency Control (MVCC) approaches in database systems, maintain multiple timestamped versions of a tuple. Once a transaction reads a tuple the database system tracks and returns the respective version eliminating lock-requests. Hence, under MVCC reads are never blocked, which leverages well the excellent read performance (high throughput, low latency) of new storage technologies. The read performance is also utilised by the read-intensive visibility and validity rules (MVCC, Snapshot Isolation) that filter the latest committed version of a tuple that a transaction can see out of the set of all tuple versions. Much more critical is the update behaviour of MVCC and Snapshot Isolation (SI) approaches, even though conceptually new versions are separate physical entities, which can be stored out-of-place thus avoiding in-place updates. Upon tuple updates, established implementations lead to multiple random writes – caused by (i) creation of the new and (ii) in-place invalidation of the old version – thus generating suboptimal access patterns for the new storage media. The combination of an append based storage manager operating with tuple granularity and snapshot isolation addresses asymmetry and in-place updates. In this paper, we highlight novel aspects of log-based storage, in multi-version database systems on new storage media. We claim that multi-versioning and append-based storage can be used to effectively address asymmetry and endurance. We identify multi-versioning as the approach to address data-placement in complex memory hierarchies. We focus on: *version handling*, (physical) *version placement*, *compression* and *collocation* of tuple versions on Flash storage and in complex memory hierarchies. We identify possible read- and cache-related optimizations.**

Keywords*—*Multi Version Concurrency Control, Snapshot Isolation, Versioning, Append Storage, Flash, Data Placement, Index.

I. INTRODUCTION

This paper is a follow-up, extended paper to our short paper published at the DBKDA 2013 [1]. We describe our Snapshot Isolation Append Storage algorithm (SIAS – [2]) in more detail, show more results of the comparison to other storage mechanisms and deliver more detailed analysis.

New storage technologies such as flash and non-volatile memories have fundamentally different characteristics compared to traditional storage such as magnetic discs. Performance and endurance of these new storage technologies highly depend on the I/O access patterns.

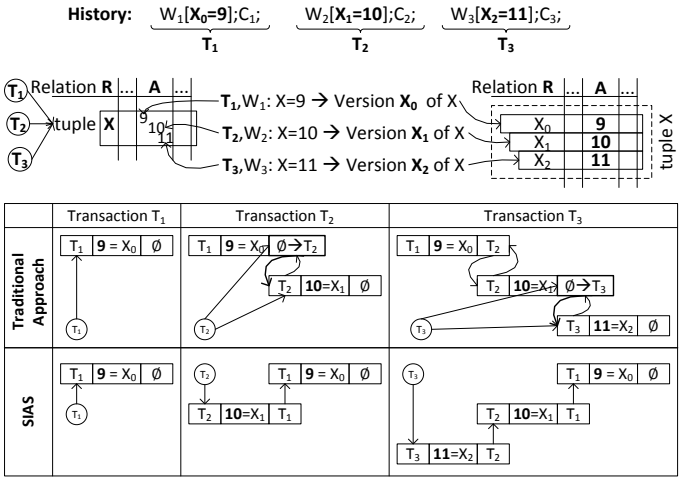


Fig. 1. Invalidation in SI and SIAS

Multi-Version approaches maintaining versions of tuples, effectively leverage some of their properties such as fast reads and low latency. Yet, asymmetry and slow in-place updates need to be addressed on architectural and algorithmic levels of the DBMS. Snapshot Isolation (SI) has been implemented in many commercial and open-source systems: Oracle, IBM DB2, PostgreSQL, Microsoft SQL Server 2005, Berkeley DB, Ingres, etc. In some systems, SI is a separate isolation level, in others used to handle serializable isolation.

Under the concept of *Append-based storage* management any newly written data is appended at the logical head of a circular append log. This way, random writes are eliminated as they get transformed into sequential writes. In-place update operations are reduced to a controlled append of the data, which is an effective mechanism to address the assymmetric performance of new storage technologies (see Section III).

In SIAS [2], we combine the multi-versioning algorithm of snapshot isolation and append storage management (with tuple granularity) on Flash. Under TPC-C workload SIAS achieves up to 4x performance improvement on Flash SSDs, a significant *write overhead reduction* (up to 52x), better *space utilization* due to denser version packing per page, better *I/O parallelism* and up to 4x lower disk I/O execution times, compared to traditional approaches. SIAS aids better *endurance*, due to the use of out-of-place writes as appends and write overhead reduction.

SIAS implicitly invalidates tuple versions by creating a successor version; thus, avoiding in-place updates. SIAS man-

ages tuple versions of a single data item as simply linked lists (chains), addressed by a virtual tuple ID (VID). Figure 1 illustrates the invalidation process in SI and SIAS. Transactions $T1$, $T2$, $T3$ update data item X in serial order. Thereafter, the relation contains three different tuple versions of data item X. The initial version X_0 of X is created by $T1$ and updated by $T2$. The *traditional approach (SI)* invalidates X_0 in-place by physically setting the invalidation timestamp and creating X_1. Analogously, $T3$ updates X_1 with the physical in-place invalidation of X_1. *SIAS* connects tuple versions using the VID where the newest tuple version is always known. Each tuple maintains a backward reference to its predecessor, which does not need to be updated in place. Hence, updating X_0 leads to the creation of X_1.

We report our work in progress on data placement and summarize key findings and the preliminary results of SIAS (published in a previous work). In this paper, we focus on novel aspects of *version handling*, (physical) *placement* and *collocation* on append-based database storage manager using flash memory as primary storage.

In Section II we present the related work. Section III provides a brief summary of the properties of flash technology. Section IV introduces the SIAS approach, aspects of *version handling*, (physical) *placement* and *collocation*. Section VI concludes the paper.

II. Related Work

SIAS organizes data item versions in simple chronologically ordered chains, which has been proposed by Chan et al. in [3] and explored by Petrov et al. in [4] and Bober et al. in [5] in combination with MVCC algorithms and special locking approaches. Petrov et al. [4], Bober et al. [5], Chan et al. [3] explore a log/append-based storage manager. The applicability of append-based database storage management approaches for novel asymmetric storage technologies has been partially addressed by Stoica et al. in [6] and Bernstein et al. in [7] using page-granularity, whereas SIAS employs tuple-granularity much like the approach proposed by Bober et al. in [5], which, however, invalidates tuples in-place. Given a page granularity the whole invalidated page is remapped and persisted at the head of the log, hence no write-overhead reduction. In tuple-granularity, multiple new tuple-versions can be packed on a new page and written together. Log storage approaches at file system level for hard disk drives have been proposed by Rosenblum in [8]. A performance comparison between different MVCC algorithms is presented by Carey et al. in [9]. Insights to the implementation details of SI in Oracle and PostgreSQL are offered by Majumdar in [10]. An alternative approach utilizing transaction-based tuple collocation has been proposed by Gottstein et al. in [11]. Similar chronological-chain version organization has been proposed in the context of update intensive analytics by Gottstein et al. in [12]. In such systems data-item versions are never deleted, instead they are propagated to other levels of the memory hierarchy such as HDDs or Flash SSDs and archived. Any logical modification operation is physically realized as an append. SIAS on the other hand provides mechanisms to couple version visibility to (logical and physical) space management. SIAS uses transactional time (all timestamps are based on a transactional counter) in contrast to timestamps that correlate to logical time (dimension). Stonebraker et al. realized the concept of TimeTravel in PostgreSQL [13]. A detailed analysis of append storage in multi-version databases on Flash is reported by Gottstein et al. in [2].

III. Flash Memories

The performance exhibited by Flash SSDs is significantly better than that of HDDs, yet Flash SSDs, are not merely a faster alternative to HDDs and just replacing them does not yield optimal performance. This section gives an extended discussion of their characteristics, as reported in [11].

(i) *asymmetric read/write performance* the read performance is significantly better than the write performance up to an order of magnitude. This is a result of the internal organization of the NAND memory, which comprises two types of structures: pages and blocks. A page (typically 4 KB) is a read and write unit. Pages are grouped into blocks of 32/128 pages (128/512KB). NAND memories support three operations: read, write, erase. Reads and writes are performed on a page-level, while erases are performed on a block level. A write is only possible to be performed on a clean (erased) block. Hence, before performing an overwrite, the whole block containing the page has to be erased, which is a time-consuming operation. Direct overwrites, as on traditional magnetic HDDs, are not possible. The respective flash memory raw latencies are: read-55s; write 500s; erase 900s. In addition, writes should be evenly spread across the whole volume. Hence, in-place updates as on HDDs are not possible, instead copy-and-write is applied.

(ii) *excellent random read throughput (I/O Operations per second – IOPS)* especially for small block sizes (as reported in [4]). Small random reads are up to hundred times faster than on an HDD. The good small block performance (4KB, 8KB) affects the present assumptions of generally larger database page sizes.

(iii) *low random write throughput*; small random writes are five to ten times slower than reads. Nonetheless, the random write throughput is an order of magnitude better than that of an HDD. Random writes are an issue not only in terms of performance but also yield long-term performance degradation due to Flash-internal fragmentation effects. Recent Flash device manufacteurs report faster random write than random read IOPS, but these figures can only be achieved by large on-device caches and do not consider sustained workload. As soon as their cache is filled, the performance of the Flash device is bound by the characteristic performance of the Flash memory.

(iv) *good sequential read/write transfer.* Sequential operations are also asymmetric. However, due to read ahead, write back and good caching the asymmetry is below 25%.

(v) *endurance issues and wear*; Flash memories are prone to wear. They only support a limited amount of erase cycles – avoiding in-place updates and reducing overwrites therefore aids longevity.

(vi) *suboptimal mixed load performance*: mixing reads/writes or random/sequential patterns leads to performance degradation.

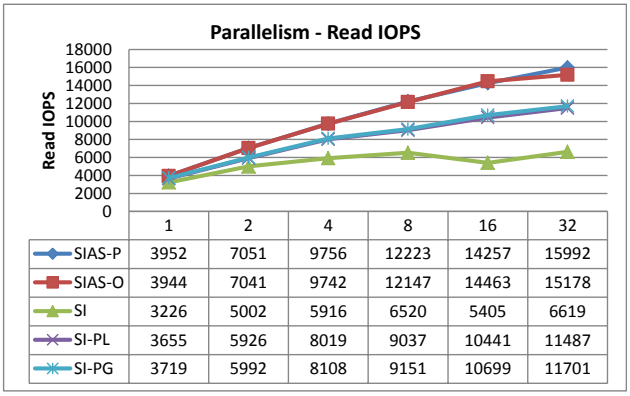
TABLE I. SIAS AND SI RESULTS ON INTEL X25-E SSD [14]

Queue Depth 1

Trace	read IOPS	write IOPS	read MB	write MB	time (sec)
SIAS-O(I)	4476	20	19713	89.96	563.675
SIAS-P(I)	4499	19	20666	89.96	587.873
SI (I)	3771	322	19901	1624	721.843
SIAS-O(II)	3947	13	11542	39.76	374.204
SIAS-P(II)	3953	13	11562	39.76	374.341
SI (II)	3656	432	11852	1395	414.869

Queue Depth 32

Trace	read I/O	write I/O	read MB	write MB	time (sec)
SIAS-O(I)	14500	66	19713	89.96	174.01
SIAS-P(I)	14642	63	20666	89.96	180.658
SI (I)	3360	264	19901	1624.9	805.193
SIAS-O(II)	15981	55	11542	39.76	92.44
SIAS-P(II)	15722	54	11562	39.76	94.128
SI (II)	11365	1338	11852	1395	133.478



	1	2	4	8	16	32
SIAS-P	3952	7051	9756	12223	14257	15992
SIAS-O	3944	7041	9742	12147	14463	15178
SI	3226	5002	5916	6520	5405	6619
SI-PL	3655	5926	8019	9037	10441	11487
SI-PG	3719	5992	8108	9151	10699	11701

Fig. 2. I/O Parallelism on Intel X25-E SSD - 60 Minute TPC-C

IV. SIAS - SNAPSHOT ISOLATION APPEND STORAGE

In this section we provide a summary of the SIAS approach [2]. SIAS manages versions as simply linked lists (chains) that are addressed by using a virtual tuple ID (VID), displayed in Figure 2. On creation of a new version it implicitly invalidates the old one resulting in an out-of-place write – implemented as a logical append – and avoiding the in-place update of the predecessor. The most recent version in the chain is known as the *entrypoint* of the chain. Without going into further details of the algorithm, the visibility is determined by accessing the entrypoint first and if it is not visible yet (long running transaction) the predecessor version is fetched using a pointer stored on the tuple version itself. In order to keep the entrypoint of each VID, SIAS employs a lightweight datastructure, where only an entry is created if the data item is comprised of more than one tuple version. SIAS is coupled to an append-based storage manager, appending in units of tuple versions and writing in granularities of pages. Only completely filled pages are appended in order to keep the packing dense, which is one of the reasons for the lower write amplification.

The example in Figure 2 shows the history of three transactions creating/updating data item X. The initial version is created by transaction T_1. Up to this point the traditional approach and SIAS create tuple version X_0. Transaction T_2 updates data item X. The traditional approach invalidates X_0 by stamping it with its own timestamp (in-place update) and creates a new version X_1 that points to X_0, analogously X_0 receives a pointer to X_1. SIAS creates the new version X_1 and stores it as the entrypoint. X_1 receives a pointer to X_0 and X_0 is left unchanged. X_1 is appended to the head of the log storage and written to the storage as soon as the page is completely filled or a arbitrary, pre-defined threshold is reached (WAL and recovery-mechanisms are left untouched). Subsequent updates proceed analogously.

Table I shows our test results with SIAS. Two traces containing all accessed and inserted tuples were recorded under PostgreSQL running TPC-C instrumented with different parameters. *Trace I* was instrumented using 5 warehouses with four hours runtime and *Trace II* using 200 warehouses and 90 minutes runtime. Both traces were fed into our database storage simulator, which generated SIAS-O/P and SI traces, containing read and written DB-pages to be used as input for the FIO benchmark, which executed them on an Intel X25-E SSD. SIAS-O is a simulation with and SIAS-P without caching of the SIAS data structures, where SI is the classic Snapshot Isolation using in-place updates on the invalidation. The conclusions of our results are:

(i) SI reads more than SIAS-O but less than SIAS-P

(ii) SI writes more gross-data than SIAS-O/P

(iii) SIAS-O/P reads with more IOPS than SI

(iv) SIAS needs less runtime than SI

(v) SIAS-O/P scales better than SI with higher parallelism.

We also conducted tests using SI and page-wise append, performing a remapping of all pages, which either appends pages local at each relation (SI-PL) or at a global append area (SI-PG) with the results displayed in Figure 2. Figure 3 illustrates the resulting write patterns using the blocktrace tool in Linux ($QD = 1$). They both achieve comparable performance in write throughput, nevertheless on subsequent read accesses the local approach has the advantage over the global approach - since the local approach makes better use of locality. This means that in the local approach pages of different relations are not interleaved as in the global approach. We found that in general both of the SI page append approaches outperform the original in-place SI by 15 to 76%, both themselfes are outperformed by SIAS-O/P by 6 to 36%. Our results empirically confirm our hypothesis that (a) appends are more suitable for Flash, (b) append granularity is crucial to performance and (c) appending in tuples and writing in pages is superior to remapping of pages. In the following sections we describe our approaches to merging of pages and physical tuple version placement as well as compression and indexing.

A. Write Amplification

One of our key benefits of the tuple based append log storage in SIAS is the significant reduction in write overhead. In our TPC-C benchmarks we observed a write reduction of up $52x$ compared to a traditional in-place update approach. The in-place update approach yields the same amount of write amplification as an append log storage manager that appends in the granularity of pages (page LbSM) – where the contents of each page are unknown.

Review the example in Figure 2, the traditional approach invalidates an old tuple version of data item X in place. This in-place update, even if it only updates a timestamp and a pointer, leads to the re-write of the page that contains the tuple version. In the example X_0 gets invalidated by transaction

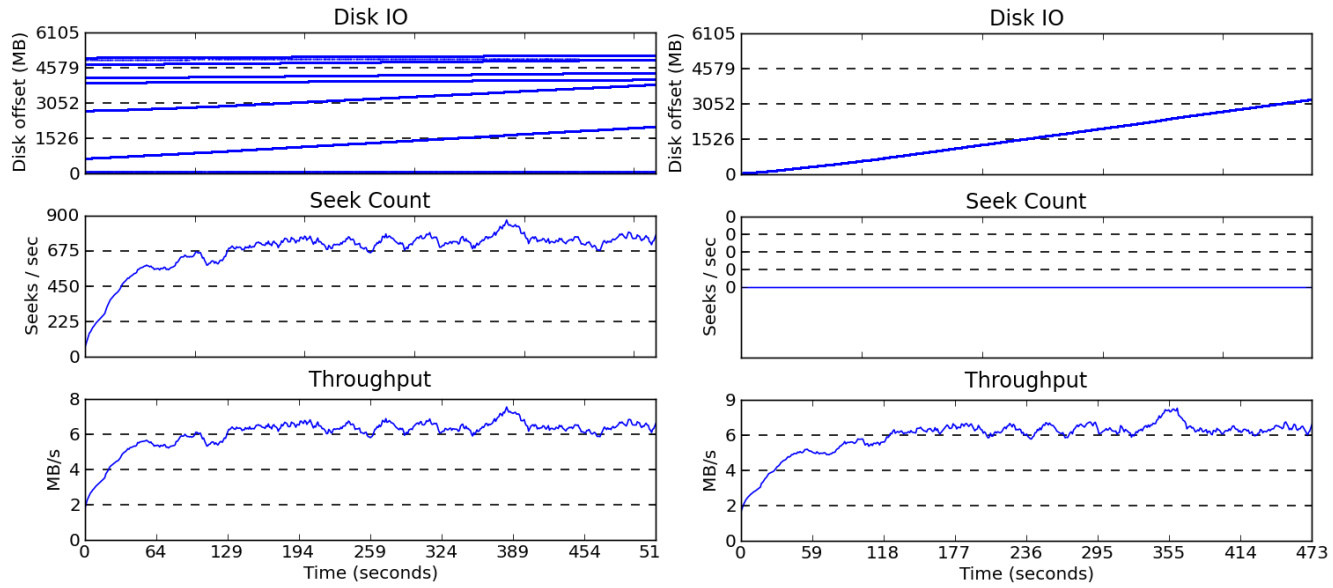


Fig. 3. Blocktrace: Left Local Append Regions (SI-PL) - Right Global Append Region (SI-PG)

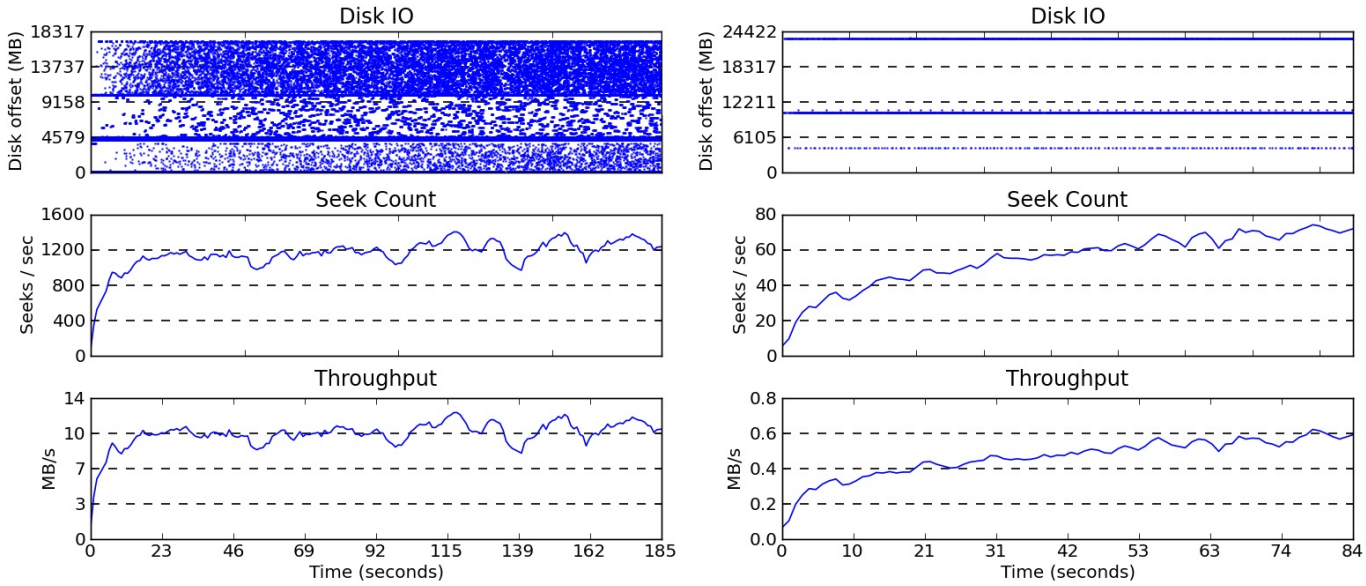


Fig. 4. Blocktrace: Write Overhead - Left In Place Update - Right SIAS

T_2, which leads to the re-write of the page that contains X_0. If the new version X_1 is stored in a different page, it also has to be written to stable storage. The update on X_0 only updates the visibility meta-information, which is necessary to determine the visible version of X, since older transactions are still able to read X_0. The traditional in-place update approach to multi-versioning, therefore, physically updates the predecessor version although the content did not change. Hence, a whole page may be re-written, which leads to a significant write amplification. One tuple version has to be inserted and in the worst case two pages have to be written. The page append storage manager transforms such in-place updates into appends, but still has to write the additional page. In SIAS this effect is alleviated by leaving the old version 'untouched'. Since the new tuple version is inserted into a new page, which is only written when it is filled (or an arbitrary threshold is reached) – this leads to the reported write reduction. The effect on the write pattern is displayed in a blocktrace diagram in Figure 4. The diagram shows the blocktrace of the default in-place update multi-versioning including its default in-place storage management and the SIAS algorithm. The workload was the resulting IO-Pattern of TPC-C trace configured with 10 clients, 200 warehouses and had a runtime of 90 minutes. The trace showed a 52 times write reduction when using SIAS (as reported in [2]). We found that the longer the trace, the higher the write reduction, which is a logical consequence since the amount of updates directly correlates with the reduction. It is also visible that the in-place approach needs more time

to complete the workload, while the append storage finishes earlier. The reason for the lower throughput in SIAS is the low amount of writes that have to be issued.

B. Merge

One key assumption of append based storage is that once data was appended it is never updated in-place. In a multi-version database old and updated versions inevitably become invisible, which leads to different tuple versions of the same data item, most likely located at different physical pages. Hence, pages *age* during runtime and contain visible and invisible tuple versions. In a production database running 24x7 it is realistic to assume that *net amount of visible tuples* on such pages is low and that an ample amount of outdated *dead* tuple versions is transferred, causing cache pollution. Once a certain threshold of dead tuples per page is reached it is beneficial to re-insert still visible tuples and mark the page as invalid. Dead tuples may be pruned or archived. Since a physical invalidation of the old page would lead to an in-place update, we suggest using a bitmap index providing a boolean value per page indicating its invalidation. The page address correlates to the position in the bitmap index, therefore, the size is reasonably small. A merge therefore includes the re-insertion of still visible tuples into a new page and the update of the bitmap index. On the re-insertion the placement of the tuples may be reconsidered (Sect. IV-C).

Space reclamation of invalidated pages is also known as garbage collection in most MVCC approaches. On flash memories, a physical erase can only be executed in erase unit granularities, hence it makes sense to apply reclamation in such granules and to make use of the *Trim* command. Pruning a single DB-page with the size smaller than an erase unit will most likely cause the FTL to create a remapping within the it's logical/physical block address table and postpones the physical erasure. This may result in unpredictable latency outliers due to fragmentation and postponed erasures [4]. Using the bitmap index, indicating deleted/merged pages (prunable), a consecutive sequence of pruned pages within an erase unit can be selected as a victim altogether. If the sequence still contains pages, which have not been merged yet, they can be merged before the reclamation.

SIAS uses data structures to guarantee the access to the most recent committed version X_v of a data item X, the *entrypoint*. If only the most recent committed version has to be re-inserted (i.e., no successor version exists), nothing but the SIAS data structure has to be updated. It is theoretically possible that the tuple version is still visible and invalidated. In this case a valid successor version to that tuple exists, which has to be re-inserted as well: Let P_m be the victim page, X_i an invalidated tuple version of data item X, where $X_i \in P_m$ and $X_v \in P_k$, $P_m \neq P_k$. X_v is the direct successor to X_i physically pointing to X_i. The merge of P_m leads to a re-insertion of X_i as X_i*, which leads to a re-insertion of X_v as X_v*, pointing to X_i*. The SIAS data structures are updated such that the most recent committed version of X know is X_i*. It is not necessary to merge P_k as well, since X_v simply becomes an orphan tuple version, which is not reachable by the SIAS data structures. Phantoms cannot occur since X_v* and X_v yield the same VID and version count. Nevertheless, it is most likely that X_i will become invisible during the merge since OLTP transactions are usually short and fast running. Further the structure is self contained on the tuples. On a crash it can be re-created by, e.g., a full sequential scan of the relation. The mapping of virtual ID, that identifies the data item, to tuple version id, which identifies the data item in a defined state in time can easily be created since each tuple version stores the VID. The existing methods of a write ahead log approach can be utilized to log changes in the SIAS datastructure.

C. Tuple Version Placement

In SIAS, each relation maintains a private append region and tuples are appended in the order they arrive at the append storage manager. Tuples of different relations are not stored into the same page and pages of different relations are not stored into the same relation regions. Appending tuple versions in the order they arrive may be suboptimal, since merged, updated and inserted tuples usually have different access frequencies. Collocation of tuples according to their access frequency can be benefitial since the net amount of actually used tuples per transferred page is higher [11]. Using temperature as a metric, often accessed tuples are hot and seldom accessed tuples are cold. The goal of tuple placement is to transfer as much hot tuples as possible with one I/O to reduce latency and to group cold tuples such that archiving and merging is efficiently backed. Visibility meta-information also contributes to access frequency, since tuples need to be checked for visibility. This creates yet another dimension upon which tuples can be related apart from the attribute values. Even if the content is not related the visibility of the tuples may be comparable.

Under the working set assumption and according to the 80/20 rule - both are the key drivers of data placement - (80% of all accesses refers to 20% of the data – as in OLTP enterprise workloads [15]) statistics can be used during an update to inherit access frequencies to the new tuple version.

In SIAS, the length of the chain describes the amount of updates to a data item (amount of tuple versions). Hence, a long chain is correlated to a frequently updated data item. A page containing frequently updated tuple versions will likely contain mostly invisible tuples after some runtime, hence simplifying the merge/reclamation process.

Version Meta Data Placement: Version metadata embodying a tuple's visibility/validity is stored on the tuple itself in existing MVCC implementations. An update creates a new version and version information of the predecessor has to be updated accordingly. SIAS benefits largely from the avoidance of the in-place invalidation. Further decoupling visibility information and raw data would be even more benefitial. Raw data becomes stale and redundancies caused by, e.g., tuples that share the same content but different visibility information are reduced or vanish completely. A structure that separately maintains all visibility information, enables accessing only needed data (payload) on Flash memory. This principle inherently deduplicates tuple data and creates a dictionary of tuple values. Visibility meta-information can be stored in a column-store oriented method, where visibility information and raw tuple data form a n:1 relation. This facilitates usage of compression and compactation techniques. A page containing

solely visibility meta-information can be used to pre-filter visible tuple versions, which subsequently can be fetched in parallel utilizing the inherent SSD parallelism, asynchronous I/O and prefetching.

Choosing the appropriate storage medium for this data is critical for performance, especially since new storage technologies change the traditional memory hierarchy augmenting it with new levels [16]. Non Volatile Memories such as Phase Change Memories seem to be a good match as they support: (i) in-place updates; (ii) fast random access (read and write); (iii) byte adressability; (iv) higher capacity than RAM. Byte addressability is important for small updates of, e.g., timestamps and to support differential updates. They still yield an inherent read/write asymmetry and are exposed to wear. A data structure within such a NVM can store pointers to raw data on flash. Our current work includes separation and placement of version information.

The SIAS data structure that stores a mapping of the virtual ID to the most recent tuple version of a single data item can be stored on such memories. In our current SIAS approach this lightweight datastructure is stored in main memory. In this way the properties of Flash memories are optimally addressed, since writes are only executed as appends and reads can be executed in parallel and smaller blocksizes. In SIAS tuple version only store stale version information, such as the creation timestamp and a pointer to the predecessor version (if the version is not the first of the data item). This also enables the usage of a multi versioned index structure that is capable of delivering the visibility decision by only accessing the index structure described in Section V.

D. Optimizations

A number of optimization techniques can be derived from observation that in append based storage a page is never updated, yet: compression, optimization for cache and scan efficiency, page layout transformation etc. Generally these facilitate analytical operations (large scans and selections) on OLTP systems supporting archival of older versions.

Compression. Most DBMS store tuples of a relation exclusively on pages allocated for that very relation. In a multi version environment, versions of tuples of that relation are stored on a page. Since all these have the same schema (record format) and differ on few attribute values at most, the traditional light-weight compression techniques (e.g., dictionary- and run-length encoding) can be applied.

Page-Layout and Read Optimizations. Since the content of a written page is immutable and only read operations can access the page, a number of optimizations can be considered. If large scans (e.g., log analysis) are frequent, cache efficiency becomes an issue, hence the respective page-layouts can be selected. Furthermore it is possible to use analytical-style page layout (e.g., PAX) for the version data and traditional slotted pages for the temporary or update intensive data such as indices. In [17] we analyse the effect of *sorted runs* in MV-DBMS' with ordered append log storage and multi-version index structures on Flash storage. It is benefitial to append in sorted runs rather than unsorted single pages and even more benefitial when it is implemented within the MV-DBMS, since the MV-DBMS is capable to use the inherent knowledge about the data. The parallelism of the Flash memories is leveraged by multiple write streams, created by the separation of append regions – each relation has its own (local) append region instead of one single (global) append region for alle relations.

V. Multi Version Index

Index structures are vital component of modern databases. Hence, their importance, especially as performance critical components, they are still a widely ignored aspect in MV-DBMS on asymmetric storage. Index structures are mostly not aware of versioned data and therefore, incapable to leverage their properties. Maintaining them on asymmetric storage becomes a critical issue.

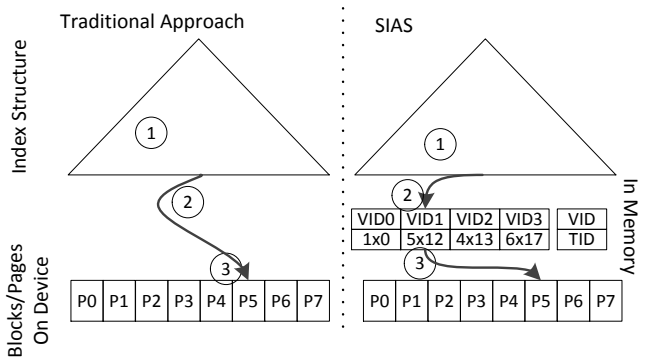


Fig. 5. Indexing: Traditional and SIAS

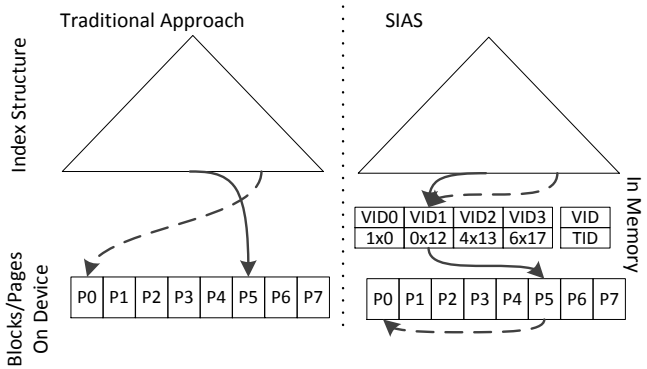


Fig. 6. Multi Version Indexing: Traditional and SIAS

Although data items exist in different tuple versions, the index addresses each version as a unique data item. This leaves the task of filtering visible versions to the rest of the MV-DBMS (e.g., executor, transaction manager). In the MV-DBMS updates of a data item lead to the out-of-place creation of a new tuple version. The index structure has to be updated (in-place) in order to index the correct tuple version that represents the data item.

The previous tuple version of a data can still be visible to some old running transactions, therefore, the index has to wait with the deletion of the pointer to the outdated version. If the index update is delayed there might be more than one tuple version of a single data item that matches the (indexed) search criteria. Hence, the index can return a data item in two different states (versions). Hence, the DBMS implementation

has to filter correct tuple versions of the data items after the access to the index. Since the visibility meta-data is stored on the tuple versions themselfes, this causes additional accesses to the I/O subsystem - even if *none* of the tuple versions is visible.

A. Index Structures in SIAS

SIAS identifies tuple versions of a single data item with a VID that is unique for all tuple versions belonging to that data item. Hence, the indexing problem can be fixed by storing the VID in the index, rather than the direct pointer to the tuple version. This gives us the benefit that indices do not have to be updated immediately when a new tuple version of a data item is created. The old entry points to the VID of the data item, which subsequently points to the most recent tuple version.

Figure 5 shows the index in the traditional approach and the SIAS algorithm. The traditional approach stores a pointer to the tuple version, treating it as a unique data item. Fetching a tuple version using such an index is comprised of 3 steps: first the index is searched using a search key. Second, assuming that a match has been found a pointer is followed. Third the tuple version is fetched and has to be checked against the visibilty criteria.

SIAS stores a pointer to the VID of the data item, which is redirected to the most recent tuple version of the data item. Fetching a tuple version is also comprised of three steps including one indirection. First, as in the traditional approach, the index structure is search using a search key. Second, assuming that a match is found, the SIAS datastructure is accessed and the pointer is followed to the most recent version. Third the tuple version is fetched and the SIAS algorithm determines the visibility.

In Figure 6, we assume that a data item exists in two tuple versions, which are both still visible. The first version of the data item is located on page $P0$ and the successor version is located on page $P5$. This case is most likely since under an LbSM approach new versions tend to be located at a position further in the log storage. In the traditional approach the index has two entries pointing to different positions on the disk. In SIAS both pointers will point to $VID1$, which stores the pointer to only the most recent version. In SIAS the index is capable of delaying updates, if now the version stored in $P0$ becomes invisible, the backwards pointer on the tuple version stored in $P5$ won't be followed and the version stored in $P5$ becomes the *stable* version of the data item. This means that the stable version is the tuple version of a data item that is committed and no running transaction is capable to read a previous version. In the traditional approach there is a tradeoff to pay, the visibility can be determined by accessing the version individually, which means that theoretically the deletion of the index entry for the old version can also be delayed but the cost of accessing the I/O storage always has to be payed.

1) Improvements on the Multi Version Index: Our current research is on the improvement of the index structure in order to be capable to answer all visibility related checks by only accessing the index structure. Hence, avoiding the access to a tuple alltogether. We have introduced an improvement that is capable of answering most of the visibility related checks by accessing the index only in [17].

VI. Conclusion and Future Work

We propose the combination of multi-version databases and append-based storage as most beneficial to exploit the distinguishing characteristics new storage technologies (Flash, NVM). When integrated they help: (i) utilise the excellent read performance low read latencies of such technologies for validity and visibility checks as well as due to the fact that readers are never blocked by writes; (ii) in addition, several types of read optimisation can be performed on LbSM level; (iii) the out-of-place update semantics resulting from the fact that upon a tuple update a new physical version is produced can be successfully utilised to reduce the expensive random writes resulting from in-place updates; (iv) existing algorithms have been revised to enable these changes.

We have prototypically implemented SIAS in PostgreSQL and validated the reported simulation results. The highest performance benefit can be achieved by the integration of the append storage principle directly into a multi-version DBMS, reducing the update granularity to a tuple-version, implementing all writes out-of-place as appends, and coupling space management to version visibility. In contrast page remapping append storage manager does not fully benefit of the new storage technology. SIAS is a Flash-friendly approach to multi-version DBMS: (i) it sequentialises the typical DBMS write patterns, and (ii) reduces the net amount of pages written. The former has direct performance implications the latter has long-term longevity implications.

In addition, SIAS introduces new aspects to data placement making it an important research area. We especially identify version archiving, selection of hot/cold tuple versions, separation of version data and version meta-data, compression and indexing as relevant research areas.

In our next steps, we focus on optimizations such as compression of tuple versions to further reduce write overhead by 'compacting' appended pages, placement of correlated tuple versions to increase cache efficiency as a 'per page clustering' approach and an efficient indexing of multi-version data using visibility meta-data separation.

Acknowledgment

This work was supported by the DFG (Deutsche Forschungsgemeinschaft) project "Flashy-DB".

References

[1] R. Gottstein, I. Petrov, and A. Buchmann, "Aspects of Append-Based Database Storage Management on Flash Memories," in *DBKDA 2013, The Fifth International Conference on Advances in Databases, Knowledge, and Data Applications*, 2013, pp. 116–120.

[2] ——, "Append storage in multi-version databases on Flash," in *BNCOD 2013, British National Conference on Databases*. Springer Berlin Heidelberg, 2013, pp. 62–76.

[3] A. Chan, S. Fox, W.-T. K. Lin, A. Nori, and D. R. Ries, "The implementation of an integrated concurrency control and recovery scheme," in *19 ACM SIGMOD Conf. on the Management of Data, Orlando FL*, Jun. 1982.

[4] I. Petrov, R. Gottstein, T. Ivanov, D. Bausch, and A. P. Buchmann, "Page size selection for OLTP databases on SSD storage," *JIDM*, vol. 2, no. 1, pp. 11–18, 2011.

[5] P. Bober and M. Carey, "On mixing queries and transactions via multiversion locking," in *Proc. IEEE CS Intl. Conf. No. 8 on Data Engineering, Tempe, AZ*, feb 1992.

[6] R. Stoica, M. Athanassoulis, R. Johnson, and A. Ailamaki, "Evaluating and repairing write performance on flash devices," in *Proc. DaMoN 2009*, P. A. Boncz and K. A. Ross, Eds., 2009, pp. 9–14.

[7] P. A. Bernstein, C. W. Reid, and S. Das, "Hyder - A transactional record manager for shared flash," in *CIDR*, 2011, pp. 9–20.

[8] M. Rosenblum, "The design and implementation of a log-structured file system," U.C., Berkeley, Report UCB/CSD 92/696, Ph.D thesis, Jun. 1992.

[9] M. J. Carey and W. A. Muhanna, "The performance of multiversion concurrency control algorithms," *ACM Trans. on Computer Sys.*, vol. 4, no. 4, p. 338, Nov. 1986.

[10] D. Majumdar, "A quick survey of multiversion concurrency algorithms," 2006. [Online]. Available: "http://forge.objectweb.org/docman/view.php/237/132/mvcc-survey.pdf"

[11] R. Gottstein, I. Petrov, and A. Buchmann, "SI-CV: Snapshot isolation with co-located versions," in *in Proc. TPC-TC*, ser. LNCS. Springer Verlag, 2012, vol. 7144, pp. 123–136.

[12] J. Krueger, C. Kim, M. Grund, N. Satish, D. Schwalb, J. C. H. Plattner, P. Dubey, and A. Zeier, "Fast updates on read-optimized databases using multi-core CPUs," in *Proceedings of the VLDB Endowment*, vol. 5, no. 1, sep 2011.

[13] M. Stonebraker, L. A. Rowe, and M. Hirohama, "The implementation of postgres," *IEEE Trans. on Knowledge and Data Eng.*, vol. 2, no. 1, p. 125, Mar. 1990.

[14] R. Gottstein, I. Petrov and A. Buchmann, "SIAS: Chaining Snapshot Isolation and Append Storage," submitted.

[15] S. T. Leutenegger and D. Dias, "A modeling study of the TPC-C benchmark," in *Proc. ACM SIGMOD Conf.*, Washington, DC, May 1993, p. 22.

[16] I. Petrov, D. Bausch, R. Gottstein, and A. Buchmann, "Data-intensive systems on evolving memory hierarchies," in *Proc. of Workshop Entwicklung energiebewusster Software (EEbS 2012), 42. GI Jahrestagung*, 2012.

[17] R. Gottstein, I. Petrov, and A. Buchmann, "Read Optimisations for Append Storage on Flash," in *IDEAS 13, 17th International Database Engineering and Applications Symposium*, 2013.

Public Healthcare and Epidemiology with Dr Warehouse

Vladimir Ivančević, Marko Knežević,
Miloš Simić, Ivan Luković
University of Novi Sad, Faculty of Technical Sciences
Novi Sad, Serbia
e-mail: dragoman@uns.ac.rs,
marko.knezevic@uns.ac.rs,
milossimicsimo@gmail.com, ivan@uns.ac.rs

Danica Mandić
Institute of Cardiovascular Diseases of Vojvodina,
Clinic of Cardiology
Sremska Kamenica, Serbia
e-mail: mandiceva88@yahoo.com

***Abstract*—The need for the systematic collection and use of epidemiological data, together with the undergoing modernization of the public healthcare system in Serbia, has motivated us to develop Dr Warehouse, an extensible intelligent software system for the collection, presentation, and analysis of data from epidemiological and public healthcare sources. The central point of the system is a data warehouse where medical data about registered disease cases and relevant demographic data are being collected. Through a web application and mobile device client, different categories of users may access data that are of interest to them, perform built-in analyses, or test their own epidemiological hypotheses. Dr Warehouse is expected to provide intuitive visualization of epidemiological data, facilitate discovery of epidemiological knowledge, and support modelling of epidemic dynamics. We discuss our motives for building such a system, the architecture of the system, our choices regarding data modelling, and the built-in functionalities. We implemented a foundation for different analyses that are expected to provide valuable insights if properly adapted and used in practice: investigation of diagnosis change over time, forecasts based on data mining, and compartmental models of disease dynamics.**

Keywords-business intelligence, public healthcare, epidemiological analysis, absenteeism, disease outbreak prediction.

I. INTRODUCTION

As a result of combining the latest advancements in business intelligence and data analysis to the domains of public healthcare and epidemiology, we present an extended version of the previously published overview of Dr Warehouse [1] – a closed source software system that supports storing of medical and epidemiological data, while offering descriptive, as well as predictive, analyses of disease cases and epidemics. There are several key issues that motivated us to develop such a solution. Despite numerous medical discoveries, lifestyle improvements, and strategies to battle epidemics, disease elimination and eradication remain as the probably most important goals in disease control [2]. Epidemiologists continue to collect outbreak data and analyse the dynamics of various ever-changing diseases in order to better understand their nature and, consequently, devise new effective countermeasures. With the proliferation of information technology, public healthcare has entered a new era with its own set of opportunities and challenges [3]. However, the increased possibilities in data collection and analysis have led to problems with the systematic treatment and use of available data [4][5]. On the other hand, the modernization of the public healthcare system in Serbia includes, among many tasks, a switch to electronic records and increase of the availability of medical information to all people involved in public healthcare. In a situation where many institutions of public healthcare conduct research separately using their in-house devised approaches, having a common point where epidemiological data could be stored could promote cooperation of these institutions and publishing of up-to-date epidemiological information to general public. As a response to these issues, we provide a potential solution in the areas of public health and epidemiology by building a software system that could help in the prevention and control of epidemics. This could be achieved by providing many procedures for different types of epidemiological research and a single data source that is tailored to the need for frequent analyses.

Within Dr Warehouse, all medical and epidemical data are stored in one such central data source, a specially designed data warehouse. Supported analyses include various data visualization techniques, statistical methods, analysis of absenteeism data, data mining algorithms, and compartmental epidemic models. Results of the analyses may be accessed through a rich web client application, which offers all of the analyses included in the system, or a mobile device client, which offers a subset of analyses primarily tailored to the needs of non-experts. Given the rapid rate of discovery of new analysis methods and epidemic models, we made the system extensible and ensured that new types of analyses and data visualization may be easily added.

The paper is organized in six sections, including Introduction. Section II offers a review of similar software systems and a comparison of their capabilities to those featured in Dr Warehouse. Section III presents our motives for building the Dr Warehouse system. In Section IV, there is an overview of the system, its architecture, featured data warehouse, and functionalities of client applications. Some

of the descriptive and predictive analyses supported by the system are presented together with sample results in Section V. Section VI includes concluding remarks and ideas for further research.

II. Related Work

There are numerous software systems for epidemiological analyses and monitoring. One group of such systems provides mostly statistical procedures that are often used in epidemiology. Open Source Epidemiologic Statistics for Public Health (OpenEpi) [6] is an example of a freely available system that may be run in a web browser [7] because it is implemented in HyperText Markup Language (HTML) and JavaScript. It focuses on statistical calculations: calculation of confidence interval and sample size, estimation of power for different types of studies, execution of various statistical tests, etc. Another free solution is WinPepi [8], which is a set of desktop applications that are similar to OpenEpi and offer many statistical procedures that are useful in epidemiology. When compared to Dr Warehouse, both OpenEpi and WinPepi are projects of a narrower scope because they ignore data storage and management. Furthermore, they put emphasis on statistics and a large number of calculation modules whose input is mostly a small set of summarized values. Unlike Dr Warehouse, they do not support data mining, visual representation of data, epidemiological maps, nor user extensions. However, the source code of OpenEpi may be directly modified to include new procedures.

The second group of epidemiological systems includes data storing and manipulation capabilities in addition to analysis procedures. Epi Info [9] is one such example of a desktop software application with a wider range of functionalities than OpenEpi and WinPepi. What sets it apart from other software systems for epidemiology is support for form creation. A user may design custom forms through an integrated editor and later use them for data entry. Besides basic and advanced statistical procedures, this system supports data import and export, as well as basic data selection and transformation. It has good data visualization capabilities and offers various types of charts, tables, and even map overlay. Its main strengths with respect to Dr Warehouse are support for form creation, direct data entry, data transformation and data import/export for various types of data sources. However, there is a conceptual difference between these two systems regarding data storage. Epi Info is a tool that may be used over any data (in the supported file or database format) and, therefore, provides transformation functions, which a user utilizes in order to prepare data for analyses. On the other hand, Dr Warehouse features a data warehouse with a fixed set of facts and dimensions, and a carefully designed ECTL process, which is automatically executed. Therefore, there is generally no need for manual data import and transformation because data preparation is done automatically. In other words, Dr Warehouse may be seen as a more specialized and more automated solution in which the data warehouse has a prominent role. Our system relies on a strong dependency between the data warehouse schema and analyses, which helps to simplify the analysis process. This, in turn, alleviates much of the burden concerning data preparation, which is usually the longest activity in analysis projects. Some of the main features of Dr Warehouse that Epi Info lacks are data mining procedures and the support for adding user extensions. We consider data mining to be an essential part of the system because, unlike most statistical procedures, it is well suited for analysing large quantities of data that are efficiently stored in a data warehouse. We may summarize this comparison by generally classifying Epi Info as a solution that offers a fixed set of analyses for any set of data attributes and Dr Warehouse as a solution that features a fixed set of data variables but an extensible set of techniques for data presentation and analysis.

The third group consists of typically web-based systems that focus on epidemiological monitoring and publicly presenting latest disease outbreak data for different regions throughout the world. They primarily rely on data from numerous Internet-related sources, which may be informal or official. HealthMap [10] provides a world map with the latest information on outbreaks by automatically collecting and integrating data mostly from several online news sources and reports from eyewitnesses and officials. There is also a mobile version of the system with similar functionalities. Another web system with a support for mobile devices is Outbreak Watch [11]. It does real-time analyses of data in social networks by evaluating keywords that are considered to be indicators of outbreaks. In this manner, the system tracks changes in the number of reports concerning relevant diseases. Google Flu Trends [12] was created as an attempt to estimate actual flu activity in various countries by analysing aggregated Google search queries that are related to flu. Since there is a relationship between an actual number of flu cases and search queries about flu, as confirmed by the overall match between the official surveillance data and the calculated estimates, this service offers near real-time results, which may help in preparing a response to a flu outbreak. Dr Warehouse is similar to these systems, as it may offer latest epidemiological data and forecasts in the form of charts, tables, and maps. In addition to supporting web access, it also features a mobile version with a selected set of services. On the other hand, the principle difference lies in the selection of data sources. The three monitoring systems use data that are available on the Internet (HealthMap and Outbreak Watch) or from web search queries (Google Flu Trends), while Dr Warehouse displays only data present in the data warehouse, which was planned to include credible data collected in healthcare institutions. However, the ECTL process in Dr Warehouse may be extended in the future to include data from public web sources.

When compared to the three aforementioned groups of epidemiological software, Dr Warehouse is a complex system that possesses traits typical of all three because: (i) it may offer any statistical procedure that has been added as an extension; (ii) data management is one of the key segments of the system; and (iii) collected data are constantly available to users via web and mobile client, which makes the system suitable for epidemiological monitoring. As a result, we consider the following two characteristics to be its major

advantages over the other solutions: (i) versatility, i.e., suitability for healthcare institutions, epidemiologists and general public; and (ii) comprehensiveness, i.e., support for data collection and storage, epidemiological analysis, data presentation, and functionality extension.

III. MOTIVATION

In addition to the prediction of epidemics and understanding of disease dynamics, we are motivated by two more specific reasons: modernization of the healthcare system in Serbia and impact of absenteeism on the economy.

As outlined in the national development strategy [13], the Serbian healthcare system is undergoing a significant transformation. Many segments of that system are being modernized to rely more on electronic records as opposed to traditional paper records. Moreover, the expected interconnection of healthcare centres would allow a better electronic access to medical data and consequently better conditions for data analyses, as in the case of the health information system (HIS) for the Serbian Ministry of Defence [14]. In such circumstances, Dr Warehouse could be integrated into the main healthcare system, which would act as a data source. After several processing steps, these data would be stored in the data warehouse within the Dr Warehouse system. Dr Warehouse has been developed also as a pilot solution that should demonstrate advantages of using a business intelligence (BI) system in the healthcare domain. It is primarily applicable within institutions that deal with disease prevention, such as institutes of public health.

Absenteeism is defined as "failing to report for scheduled work" [15]. High absenteeism has negative impact not only on colleagues and superiors, who must cope with greater workloads, but also on the profit. According to the 2009 research by the Chartered Institute of Personnel and Development (CIPD) from Great Britain [16], the most important reasons for the short-term work absence (up to four weeks) are: colds, influenza, stomach problems, headaches, migraines, injuries of the muscular and skeletal system, and pain in the lower back part. Most of these conditions, which are also a major health problem in Serbia, are preventable non-communicable diseases (NCDs). However, there is no adequate prevention and control of NCDs in Serbia [17]. We hope that, by using Dr Warehouse, valuable absenteeism patterns could be uncovered.

There are three primary groups of users who might benefit from the developed system: employees in public healthcare institutions, researchers in epidemiology, and non-experts interested in epidemiological information. Users in public healthcare institutions that are dealing with epidemiological data could utilize our software system, which is specially tailored to the epidemiological domain, instead of relying on solutions that are intended for generic statistical analyses. An expected advantage of having a domain-specific system would be an increase in user productivity. Large amounts of data that are typical of modern HISs may be well utilized owing to the well-tried approach incorporated into our system – a data warehouse for data storing and data mining for efficient analyses. The main system load may be reduced by running analyses primarily on data stored in Dr Warehouse. The second group of users are scientists whose research is related to epidemiology. By utilizing Dr Warehouse, they may create, test, and improve epidemic models through adding, running, and modifying new extensions. New visualization techniques for epidemiological data may also be employed and evaluated. Furthermore, the system may also target users who are not medical experts but are interested in latest disease trends, forecasts, or results of some specific analysis.

Owing to the adverse health of the population in Serbia and the "white space" in terms of medical services aimed at predicting disease occurrence, our decision to develop a system that would allow the use of BI technologies in such a context should be both socially and economically justified.

IV. SYSTEM OVERVIEW

In this section, we present the system and give an overview of its architecture and functionalities. Public resources concerning the system are available at [18]. The featured data warehouse, which represents a foundation for data analyses, is explained in more detail. We also elaborate on the featured web application, mobile application, and built-in support for adding new functionalities.

A. System Architecture

There are four principal components in the system: (i) database server, (ii) application server, (iii) web client application, and (iv) mobile device client application. The overview of the system is given in Fig. 1.

The database server is depicted as a rounded rectangle titled *SQL Server* (in the left portion of Fig. 1). It consists of a relational database management system (*SQL Server RDBMS*), which hosts a data warehouse containing epidemiological and medical data; services for data analyses that focus on data mining and OLAP cube analytics (*SQL Server Analysis Services*); and data integration services (*SQL Server Integration Services*) for data extraction, transformation, and loading (the ETL process) from the supported types of data sources (Excel files, various relational databases, data files, or some other external sources).

The application server is depicted as a rounded rectangle titled *Dr. Warehouse Server* (in the central portion of Fig. 1). It acts as an intermediary between the database server and the two client applications. This component communicates with the database server using the *ADOMD.NET* subcomponent, which is responsible for providing access to analytical data sources. On the other hand, web services (*WCF Services*) are used to exchange information between the application server and the two client applications. New functionalities (*Extensions*) may be added to the application server using the *MEF* subcomponent.

The web client application, which is depicted in Fig. 1 as a rounded rectangle titled *Silverlight Client Application*, may be extended in the similar manner using its own *MEF* subcomponent. It also supports reading of Serbian identity cards (ID cards) using the *Smart Card Reader* module. The mobile device client is depicted in Fig. 1 as a rounded rectangle titled *Windows Phone 7*.

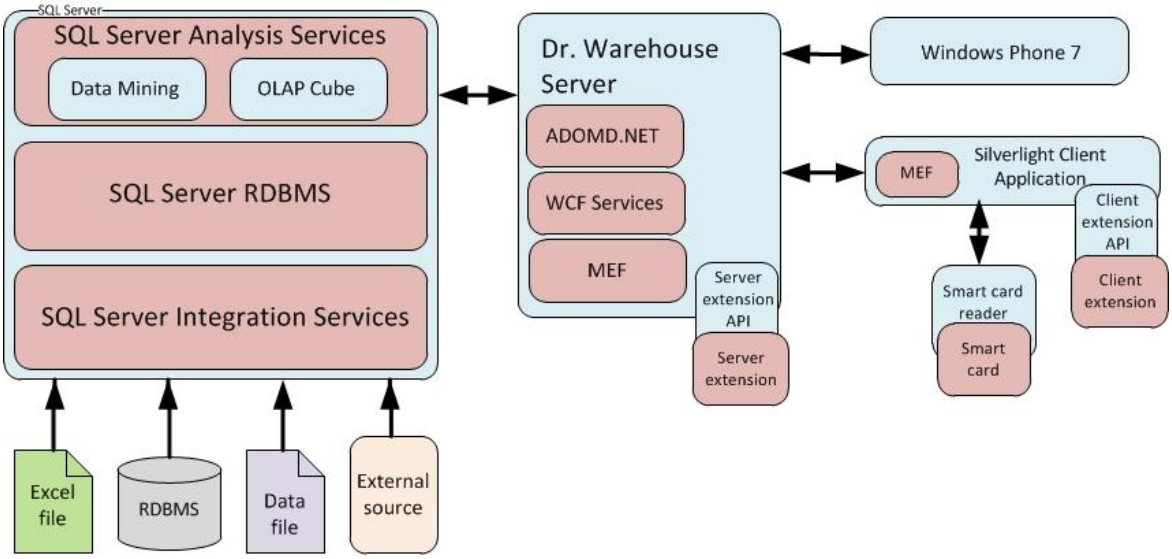


Figure 1. System overview.

The system may fit into existing HISs and provide various services to other similar solutions. This architecture allows the possibility of having the database server and application server reside at different physical locations. Furthermore, in order to increase the scalability and performance of the system, the data mining and analysis services (currently implemented using Microsoft SQL Server Analysis Services [19]) may be located separately from the database server. In future versions of the system, the architecture may be extended to include terminals that would be publicly available and offer a set of functionalities similar to those in the existing web client application.

B. Data Warehouse

The data warehouse is modelled using a star schema, which consists of eight dimensions, two of which are role-playing dimensions, and one fact table (Fig. 2). The fact table keeps track of events that lead to absenteeism, disease occurrences and time measured in days that person spent away from duty or workplace. Each dimension represents the context of disease occurrence and absence. Therefore, we can observe these events in the context of time (when an event occurred or ended), gender of the person involved, place where it happened, person's profession, data source, absence cause, person's age, and diagnosis that was established.

Dimensions concerning diagnosis, place, and time have several hierarchical levels modelled as a fully denormalized structure, which enables multi-level classification of factual data. In the time dimension, we have two hierarchies: one defined as calendar year, quarter, month, and day, and the other one as calendar year, week, and day.

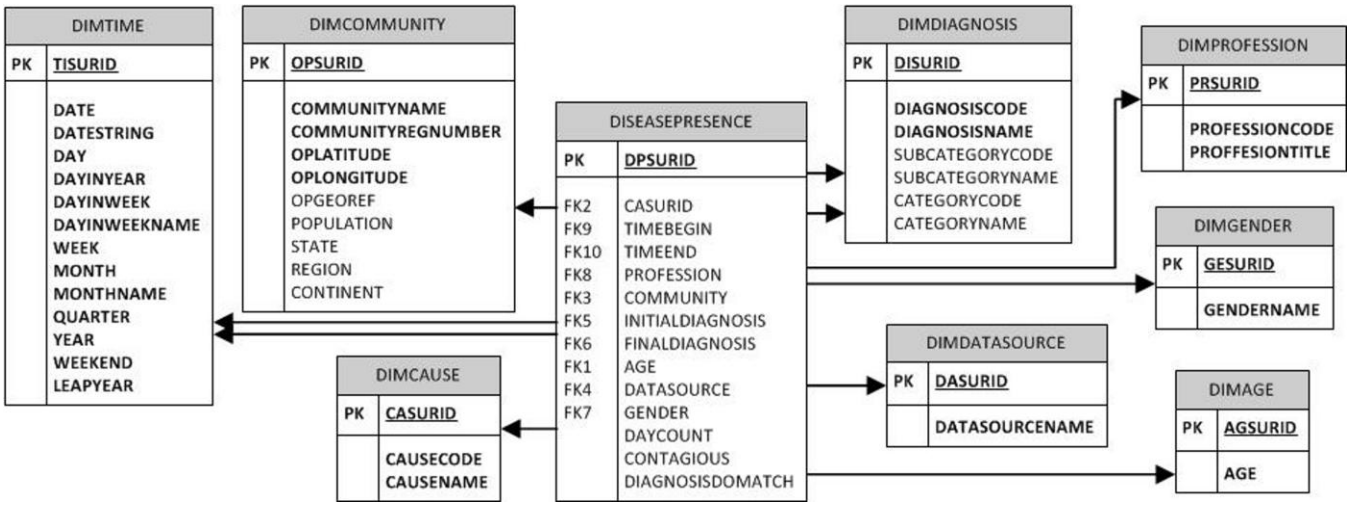


Figure 2. The star schema of the data warehouse.

The diagnosis dimension has three levels of hierarchy for diagnosis, disease subcategory, and disease category. Within the community (place) dimension, there are four levels of hierarchy for community, state, region, and continent.

Although the normalization of our schema would remove redundant data, which in turn would make the schema easier to maintain and change, our initial considerations of the schema type led us to choose the star schema. Denormalization, which is typical for the star schema, helped us to reduce the number of foreign keys and to reduce the query execution time. As the system was designed to be used by a wide variety of users, ease of use was one of our priorities. For end users, the star schema is more comprehensible than snowflake schema and less complex queries are needed to satisfy their information needs. Since this is a pilot project, advanced cost-benefit analysis of normalizing our star schema into the showflake schema is a matter of our future work. The unavailability of a larger and more complex absenteeism data set was a major reason for simplifying the initial schema design and focusing on the aforementioned fact and dimensions.

The data warehouse was implemented using Microsoft SQL Server 2008 [20]. It includes the following dimensions: *DimCause*, *DimDiagnosis*, *DimGender*, *DimProfession*, *DimCommunity*, *DimDataSource*, *DimTime*, and *DimAge*. *DiseasePresence* is the only fact table in the system. Each of these tables contains a surrogate primary key, which allows us to deal with changes in natural key in a more convenient way and track slowly changing dimensions.

Taking into consideration that the data in the system are expected to reflect the actual state of the health of a population, it is necessary to support acquisition and integration of medical data from multiple sources. We developed a solution within Microsoft Integration Services [21], which allows us to extract, clean, transform, and load (ECTL) the necessary data. The ECTL process is divided in six SQL Server Integration Services (SSIS) packages each of which covers control and data flow between sources (Excel files or databases) and corresponding fact or dimension table. Furthermore, for each package, set of actions is specified within SQL Server Agent [22] jobs. Those actions involve preparing source files, configuring connections to data flow sources and destination, executing the packages and backing up old source files. Moreover, the execution of SSIS packages is logged and completion status is emailed to an administrator.

We perform incremental extraction, i.e., we consider only data that were added to the HIS of a public health institute or uploaded to application sever after the previous extraction. Extracted data serve as an input for a series of transformations in which we detect and eliminate errors and inconsistencies: (i) different domains of semantically equivalent attributes (as in the case of the attribute *GenderName*); (ii) different encodings of textual data (*ProfessionTitle*); (iii) different granularity of semantically equivalent attributes (*DiagnosisCode*). Diagnosis codes that are used in the source HIS are shorter versions of the codes that are featured in the 10th revision of International Classification of Diseases (ICD 10) [23]. We created a transformation that relies on regular expressions to resolve this issue. In this manner, we extended disease information with the disease name, subcategory and category. Dimensions *DimGender*, *DimTime*, and *DimAge*, which are static dimensions, are not extracted from data source. They are created within the context of the data warehouse and their records are either loaded manually or generated by a custom procedure.

At the moment, there is only support for data insertion. Since the data set in the current version of the system is only a sample taken from a HIS, we decided to keep all data in the data warehouse, while leaving the implementation of a deletion policy for obsolete data and data that have little or no impact on the system output, to be included in the future version.

In order to meet the needs for efficient and flexible consumption of valuable information produced by the system, we developed an online analytical processing (OLAP) database, which contains rich metadata. The OLAP cube makes our data organized in a way that facilitates non-predetermined queries for aggregated information. As we used Kimball Method [24] to implement the dimensional model in the relational database, the OLAP design step was a straightforward translation from the existing design. The relational database serves as the permanent storage of the cleaned and conformed data, and feeds data to the OLAP database. Data mining structures and models are stored in the third database, which, together with the OLAP database, resides at the Analysis Server – the primary query server in the system.

C. Web Application

The majority of the functionalities that are available to expert users are incorporated into a web application, which is implemented in Microsoft Silverlight [25]. The communication between the application server and the web client is done via web services using Microsoft Windows Communication Foundation (WCF) [26]. At present, the client application possesses functionalities concerning: access to medical records stored in the data warehouse; upload of data files containing medical records; access to the data cube and use of some of the cube's advanced analytical operations; execution of advanced analyses and forecasts, as well as result retrieval; upload of extensions; and their invocation.

The contents of the web application are organized as a set of Silverlight pages, where each page groups a number of similar functionalities. Within the *Home* page, users may access a chart about the most common causes of absenteeism for the current month. Moreover, in order to fulfil the needs of more experienced users accustomed to traditional reports, we created various operational reports within the server-based reporting platform, SQL Server Reporting Services (SSRS) [27]. Besides data sheets, these reports include rich data visualization in form of 3D charts. Users may access reports on-demand through a web browser. After they run a report, they can export it to another format, such as Excel spreadsheet or PDF. The functionalities of other pages are presented in the remainder of the subsection.

1) General Predictions

The *Analysis* page contains functionalities regarding the execution of advanced analysis and forecasts that identify the most probable diseases (or causes of work absence) and the most frequent diagnosis mismatches. Through this page, a user is able to generate basic predictions concerning a selected subpopulation for a particular quarter of a year. The subpopulation may be specified by selecting an age group, gender, and municipality (Fig. 3). More information about these predictions may be found in Section IV-C.

2) Personalized Predictions

The *What about me?* page is a location from which it is possible to generate and retrieve results of the personalized predictions concerning the most probable diseases (or causes of work absence). In order to generate these predictions, a user is required to insert his or her ID card into the attached smart card reader and a report is automatically generated.

Card data are read with the help of the Čelik API [28], which is primarily intended for integration of ID cards into business systems. In order to access the smart card reader from a web browser, we had to enable trusted applications to run inside the browser. Moreover, as soon as health smart cards become publicly available in Serbia, we intend to adapt the system to allow data reading from these cards as well.

3) Analytical Operations

The execution of analytical operations and access to historical data is provided within the *Health Reports* page. Users may perform operations such as dice and slice in order to analytically process the available data. The *PivotViewer* control [29], which is an integral part of the page, helps users to interact with thousands of items at once and see trends and patterns that would be hidden when looking at one item at a time. Fig. 4 shows the *PivotViewer* control in which green squares represent diagnoses and the numbers of their occurrences among males of a certain age. The user is able to zoom in and select squares in order to retrieve more information on the corresponding diagnosis such as description and number of occurrences.

4) Upload of New Data and Functionalities

The *Upload* page offers functionalities regarding uploading of server and client extensions. Within the *Extensions* page users may activate and run uploaded extensions. Furthermore, through the *Upload* page users may upload Excel files containing medical records. As most HISs support exporting data in the form of Excel files, we considered it to be most suitable format for the task. Every uploaded file is stored with a unique name, which prevents file name collisions. In order to minimize the impact of the intense ECTL activity on user experience, we scheduled a server job that executes SSIS package every day at 12:00 pm.

However, we are going to reconsider this decision once we gather concrete data on the frequency of data upload and user activity within the web application. The first step in the job is reserved for preparing the uploaded file for the ECTL process and the last step for backing up the file. All steps within the job are repeated until there are no more files in a directory on the server where uploaded files are stored.

5) Diagnosis Discovery Support

The *Symptoms Checker* page is part of our more recent research, which resulted in the functionality that supports medical diagnosis discovery. The diagnostic process is based on symptom matching in a way that is sensitive to the diagnoses that are dominant in the population to which the patient/user belongs. This functionality, which is presented in more detail in [30], utilizes data sets presented in Section IV-A, and a two-phase algorithm that is based on the differential diagnosis method from medical diagnostics and predictive models for disease occurrence in a subpopulation.

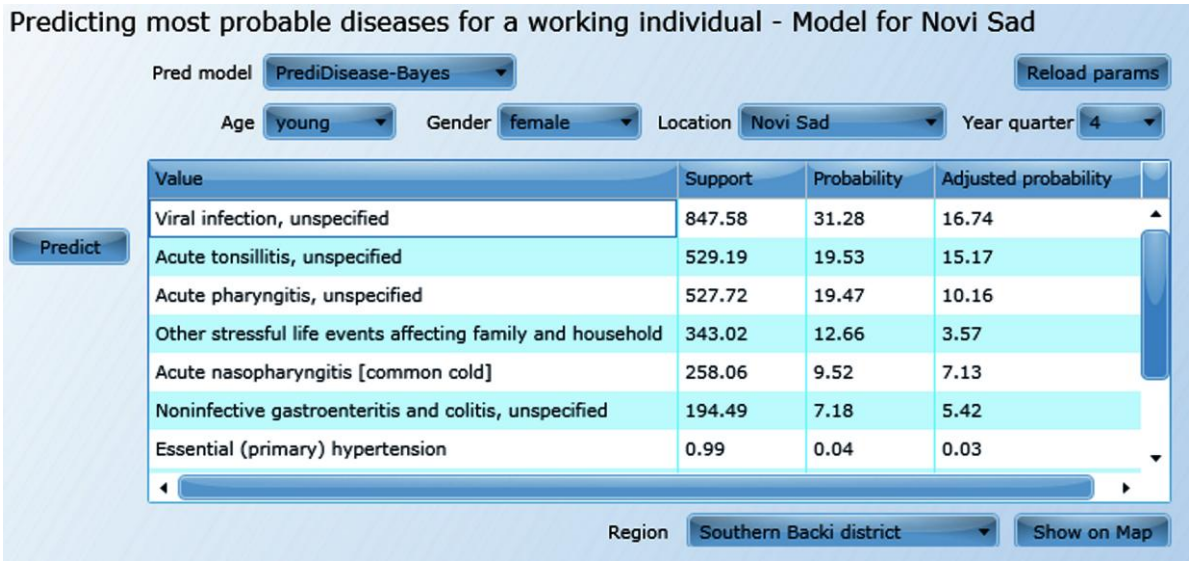


Figure 3. Section from the Analysis page in the web client.

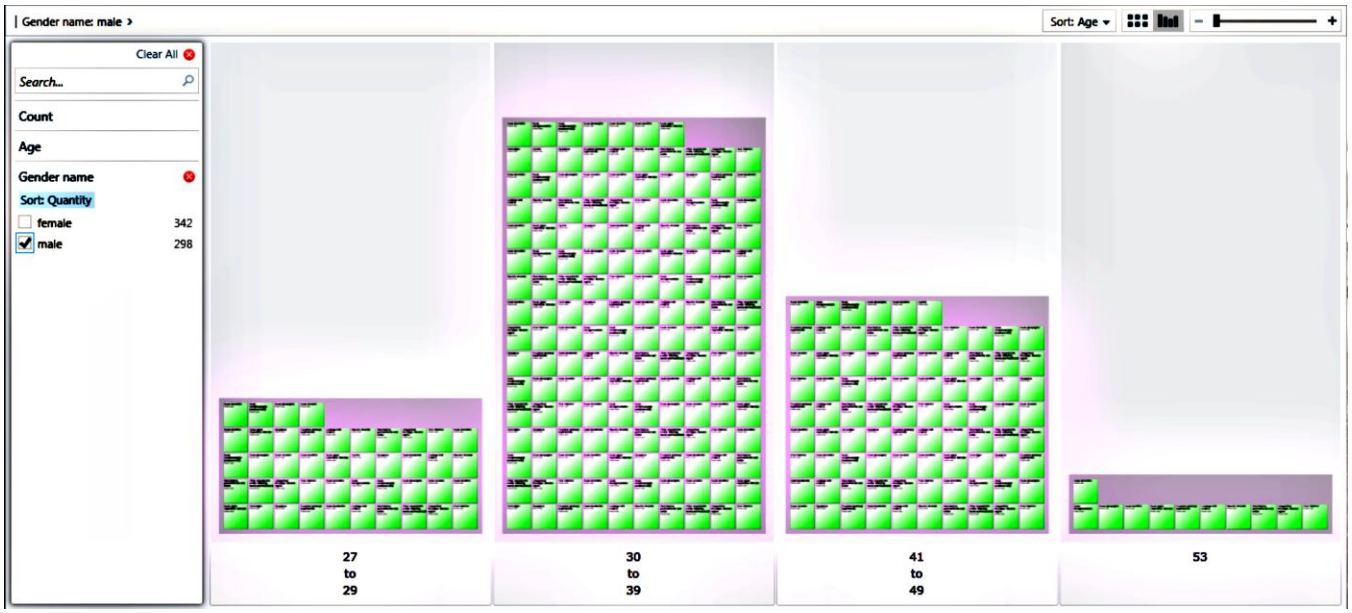


Figure 4. Data visualization using PivotViewer.

D. *Mobile Application*

Dr Warehouse Mobile is a mobile client for the Dr Warehouse system. It is an application designed for smartphones and implemented only for the Microsoft Windows Phone platform [31]. However, porting the mobile application to other platforms such as Android and iOS is a matter of future work. Given the fact that smartphones are widely used, we have chosen to offer a mobile application as a way of integrating the collected medical information and predictions from Dr Warehouse into regular activities of potential users.

The mobile application has functionalities similar to those offered through the web client. However, it has a narrower set of features that are customized for mobile users. Its functionalities are organized into several segments: *Login Process*, *Registration Process*, *Current Position*, *Position Information*, *Personal Information*, and *Symptoms*. In order to exchange data between the mobile client and application server, we developed WCF web services tailored for the mobile client. The methods for determining the common diseases and potential diagnoses are the slightly modified versions of the corresponding methods available through the web application.

Once started, the mobile application presents a login screen (*Login Process*) with the option to navigate to the user registration form. When registering in the system (*Registration Process*), a user provides personal information such as age, gender, profession, and place of residence. The provided information is used when performing a personalized analysis similar to that featured in the *What about me?* page in the web client.

After a successful login, the user is provided with the position (*Current Position*) obtained via the GPS receiver of the phone and Microsoft TerraService [32]. With this information, the user may inquire about the most common diseases at the current location (*Position Information*). For this purpose, the client issues an asynchronous call to a web service. Once the response is received, the user is informed via the toast pop-up notification and information about the most common diseases is shown on a map. Moreover, the same information, in the textual or chart form, may be obtained for any location that is designated by the user.

In the similar manner, the user may retrieve a list of diseases that are most probable for the subpopulation to which the user belongs, where the subpopulation is determined using the personal information provided during the registration, or for any other specified subpopulation, where the user needs to specify the required information: age, gender, profession, and location. The scenario of obtaining prediction according to the personal information (*Personal Information*) is illustrated in Fig. 5.

The last segment of the mobile application is devoted to providing potential diagnoses for a list of symptoms exhibited by the user (*Symptoms*). The user may create a personal symptom list by adding observed symptoms from a list of the common symptoms, which is regularly updated from the server, or by manually entering the name of a less common symptom. After the symptom list is submitted for the evaluation, the list of potential diagnoses is retrieved from the server. For each diagnosis, there is additional information such as risk factors, treatments, and aliases (Fig. 6). Furthermore, the user may access a separate description of a potential diagnosis. For each provided piece of information about a diagnosis, there is a hyperlink to a relevant resource with a more detailed description.

Figure 5. Personalized predictions within the mobile client.

Figure 6. Description of a potential diagnosis within the mobile client.

E. *Extensibility*

New functionalities may be added to the system in the form of extensions. The support for extensibility was implemented using Managed Extensibility Framework (MEF) [33]. A user may upload an extension, which then becomes immediately available for use without a need to restart the system. There are two types of extensions: (web) client extensions and (application) server extensions. Both may be uploaded to the application server through the web client. A web client extension is automatically downloaded from the application server to a web client machine, where it is then executed. This is done upon the first invocation of the extension at the client side. Such extension is actually a Silverlight web page that is embedded within the *Extensions* page in the web application. It is generally expected to act as a user interface to the built-in or user-added (via server extensions) queries and analyses. On the other hand, server extensions reside on the application server, where they are also executed upon the invocation initiated at the client side. These extensions are functions generally responsible for data operations, analyses, and epidemic models.

V. FEATURED EPIDEMIOLOGICAL ANALYSES

In this section, we present three types of epidemiological analyses that are available in Dr Warehouse: an analysis about the difference between the initial and final diagnosis during work absence, epidemiological forecasts that rely on data mining and epidemiological forecasts that rely on compartmental models. In addition to describing a data set that was used, we offer exemplary results of these analyses. These results primarily illustrate what kind of valuable information may be obtained from Dr Warehouse. Their validity is tightly coupled with the quality of available epidemiological data and the precise tuning of procedure parameters.

A. *Data – Sample, Quality, and Security*

Data set used in the testing of the system during the development was acquired from the HIS of The Institute of Public Health of Vojvodina in Novi Sad, Serbia. The obtained sample (an excerpt is featured in Fig. 7) has approximately 8,500 records about workplace absences that ended in 2009. It contains depersonalized information including: gender (represented by the variable *pol*), age (*starost*), municipality code (*opstina*), absence cause (*uzrok*), start (*prvidan*) and end date (*krajdan*) of absence, disease codes for initial (*pdijag*) and final (*zdijag*) diagnosis, and business activity code (*delatn*) of a person involved.

Gender is represented by numbers 1 and 2 referring to the male or female respectively. Business activity code is represented by a five-digit code indicating sector, division, branch and group of a business activity in accordance with the classification of activities as defined by the corresponding law of the Republic of Serbia. Municipality code is a unique identifier of the municipality in which an absence was recorded.

The cause of the absence is denoted by numbers from 1 to 12 that respectively correspond to: disease, isolation, accompanying sick person, maintenance of pregnancy, tissue and organ donor, injury at workplace, injury outside of workplace, occupational disease, nursing a child under 3 years, nursing a child over 3 years, care of other sick person, and maternity leave. Initial and final diagnosis codes are obtained by reducing the appropriate diagnosis codes defined by the 10th revision of International Classification of Diseases (ICD 10) to four characters. Codebooks of diseases, business activities, causes and municipalities may be gathered from official Internet sites of organizations that are responsible for their maintenance and distribution.

	pol	starost	delatn	opstina	uzrok	pdijag	zdijag	prvidan	krajdan
1	2	52	80220	2690	1	M543	M543	19-Jan-2009	20-Jan-2009
2	2	48	92522	2690	1	M539	M539	12-Jan-2009	23-Jan-2009
3	1	40	51340	1250	1	J42X	J42X	23-Dec-2008	12-Jan-2009
4	1	23	51530	2690	1	M549	M549	20-Jan-2009	21-Jan-2009
5	2	40	85321	1250	1	J42X	J42X	12-Jan-2009	29-Jan-2009
6	1	30	34300	2690	9	Z637	Z637	05-Jan-2009	16-Jan-2009
7	2	30	01110	1250	10	Z637	Z637	26-Jan-2009	26-Jan-2009
8	2	30	01110	1250	10	Z637	Z637	12-Jan-2009	12-Jan-2009
9	2	30	01110	1250	10	Z637	Z637	19-Jan-2009	21-Jan-2009
10	2	26	01110	1250	10	Z637	Z637	30-Jan-2009	30-Jan-2009

Figure 7. Excerpt from a data set used in the generation of predictions.

Credibility of the data depends largely on the credibility of data sources. Therefore, we rely on sources that can guarantee the integrity and validity of provided data.

Within the system, we provided different ways of presenting and making use of existing data. Dr Warehouse is not only conducive to making decisions for medical experts by means of its data rich reports, but also favourable to general population because of its easy to use components. In such way we tended to satisfy some of the secondary data quality criteria such as reliability, credibility, usefulness, added value, ease to use, and accessibility. However, the conception and measurements according to these criteria are established on primary or secondary quality properties which are mainly assessed by subjective methods [34]. The denormalization of the data warehouse schema, which is described in Section III-B, affects some of the properties corresponding to the primary data quality criteria such as efficient use of storage and response time. Although it entails the generation of redundant data, we consider it beneficial because it optimizes read performance and makes the schema more comprehensible to users.

Security in software systems that contain medical data is generally one of the top concerns of software designers. In the current version of Dr Warehouse, the featured data warehouse supports storage only of depersonalized medical records. In such design, there are generally no standard issues with patient privacy and confidentiality of patient health records. Moreover, Dr Warehouse was developed to give wide access to epidemiological data to different categories of users. However, future versions may support distinct user roles that include specific functionalities and access to different portions of contained data. The future access control is expected to follow the role-based access control (RBAC) model [35].

In the more recent development of the system, we extended our solution with a data set acquired from Freebase, an open repository of structured data [36]. We were able to downloaded JavaScript Object Notation (JSON) [37] files that contain structured information on findings consistent with diagnoses. After parsing those files, we obtained records about diagnoses, corresponding symptoms, risk factors, disease causes, treatments, and medical specialties. Those records were stored in a specially designed relational database which is presented in more detail in [30].

B. Changes in Diagnosis

The situation when the final diagnosis in an absence case differs from the initial one may be of special interest to medical experts. Analysing cases in which complications lead to a change in diagnosis or finding often misdiagnosed cases could help experts to devise strategies for the prevention or control of such situations.

For these reasons, we added support for the comparison between the initial and final diagnosis associated with a single workplace absence. For each pair of the initial (*L*) and final (*R*) diagnosis that appears in the available data set, we automatically calculate the number of matching absence cases (*Support*), their share in the whole data set (*Support %*), the percentage of cases with the initial diagnosis *L* that also feature the final diagnosis *R* (*Prob_LR*), and the percentage of cases with the final diagnosis *R* that also feature the initial diagnosis *L* (*Prob-RL*). A user may choose to view only the specified number of the most frequent diagnosis changes, and further select either the initial or final diagnosis to investigate which diagnosis changes are associated with the selected diagnosis.

An excerpt from the results of an analysis about diagnosis change is given in Fig. 8. The majority of changes between the initial and final diagnosis is observed when the supervision of normal pregnancy was substituted with the health supervision and care of other healthy infant and child. This scenario happened in 258 cases, which make up 99.61% of normal pregnancy cases. A similar trend is noticed when the initial diagnosis was officially recorded as the supervision of normal first pregnancy (50 cases). Such diagnosis changes could be considered trivial as they are numerous and do not have much informational value.

On the other hand, the diagnosis changes with lower support value could better demonstrate in what ways a diagnosis may change. In 43 cases with an unspecified abdominal pain as the initial diagnosis, there were six different outcomes (Fig. 9). In the majority of such cases (83.72%), the diagnosis remained the same. The most common change was the one to dyspepsia, which happened in 3 cases (6.98%). The remaining cases further illustrate how a set of unrelated diagnoses, such as cholelithiasis, lumbago, haemorrhoids, and gastroduodenitis, could substitute an imprecise initial diagnosis. As a result, the discovered information might indicate for which common alternative diagnoses a more thorough test could be administered in order to reduce the chance of making a wrong diagnosis.

C. Forecasts based on Data Mining

In Dr Warehouse, we utilize three classification algorithms that are supported by Microsoft SQL Server 2008 R2 Analysis Services: decision trees, naive Bayes, and neural network classifier. These classifiers are trained to estimate the individual share of each disease in all work absences attributed to the 15 most common diseases, as determined by examining the available data set, for a selected year quarter and subpopulation, as defined by age group and gender, in a selected municipality.

Description	Support %	Support	Prob-LR	Prob-RL
Supervision of normal pregnancy, unspecified -> Health su	3.07	258	99.61	83.77
Supervision of normal first pregnancy -> Health supervisio	0.59	50	94.34	16.23
Viral infection, unspecified -> Acute bronchitis, unspecified	0.06	5	1.31	2.7
Acute pharyngitis, unspecified -> Acute bronchitis, unspeci	0.05	4	0.83	2.16
Other and unspecified abdominal pain -> Dyspepsia	0.04	3	6.98	3.37
Angina pectoris, unspecified -> Presence of coronary angic	0.02	2	6.06	18.18

Figure 8. Some of the most frequent changes in diagnosis.

Description	Support %	Support	Prob-LR	Prob-RL
Other and unspecified abdominal pain -> Other and unspecified abdominal p	0.43	36	83.72	100
Other and unspecified abdominal pain -> Dyspepsia	0.04	3	6.98	3.37
Other and unspecified abdominal pain -> Other cholelithiasis	0.01	1	2.33	4.76
Other and unspecified abdominal pain -> Lumbago with sciatica	0.01	1	2.33	0.39
Other and unspecified abdominal pain -> Unspecified haemorrhoids without	0.01	1	2.33	4.17
Other and unspecified abdominal pain -> Gastroduodenitis, unspecified	0.01	1	2.33	1.39

Figure 9. Cases with an unspecified abdominal pain as the initial diagnosis.

In this manner, we may form coarse predictions of the distribution of the most common diseases in a selected subpopulation. In Fig. 10, we give a set of predictions for male employees in the city of Novi Sad who are between 40 and 61 years old. This example demonstrates how a share of some common diseases in that subpopulation may change throughout a year. These estimates are generated using the naive Bayes classification algorithm for Novi Sad.

Predicted shares indicate that essential hypertension, dorsalgia (thoracic region), and lumbago with sciatica may be causes of a larger percentage of absence in quarters 2 and 3 (spring and summer), while their share substantially decreases during quarters 1 and 4 (winter and autumn). On the other hand, viral infection is most responsible for absences in quarter 4 (autumn).

D. Forecasts based on Compartmental Models

Compartmental models are a group of epidemic models that are used to predict dynamics of an epidemic by dividing an analysed population into several compartments (subpopulations) and calculating the changes in compartment sizes given some initial conditions [38][39][40]. These conditions include sizes of compartments (generally expressed as percentages of a whole population) at a single moment in time.

Population compartments correspond to susceptible, infectious, recovered, or some other group of individuals in a population. Furthermore, there are disease-related parameters that are needed in the calculation of changes in compartments sizes: contact rate, recovery rate, birth/death rate, etc. Different models from this family feature different compartments and may be used to obtain forecasts for different diseases. The actual spread of a disease (transition of individuals between different compartments) is modelled by a system of differential equations.

Compartmental models may be used to predict epidemics. However, the modelling of a disease offers additional benefits. Compartmental models are typically used to analyse the state of equilibrium for a particular disease, i.e., the point when there are practically no more changes in the size of featured compartments. Moreover, the information from such models may be applied to control or even prevent outbreaks by using it to define an adequate vaccination strategy.

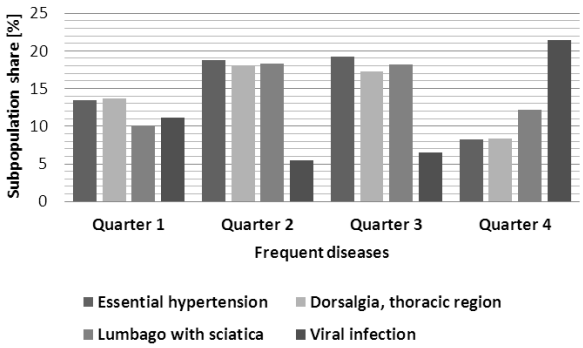


Figure 10. Example of percentage disease shares for male employees in Novi Sad aged between 40 and 61 years, as predicted using a naive Bayes classifier.

As an example of how Dr Warehouse may support standard epidemic models, we implemented five compartmental models based on the information from [41]. With these models, it is possible to describe some common diseases such as influenza, measles, and sexually transmitted diseases. Moreover, they describe with different levels of detail the nature of infection and potential immunity, which is suitable for a broad range of diseases. Both the short-term and long-term forecasts may be made owing to possibility to include basic demographic processes in these models.

The implemented models are added to the system in the form of extensions. For each model, there is a separate client extension. Each client extension is a Silverlight page within the web application. These pages are used for actions such as setting parameters, invoking model execution, and presenting results. In addition to client extensions, there is a single server extension for all implemented models. This extension contains functions that are responsible for numerically solving systems of equations which correspond to compartmental models. Our implementation approximates the solution by using the 4th order Runge-Kutta method for solving a system of ordinary differential equations. It is invoked from a client extension and provides results for the client side.

Various compartmental models may be implemented in a similar manner. The major differences in the implementation would be a change in the set of differential equations that model a disease and addition of new parameters or compartments. By adding the support for several different compartmental models in the form of system extensions, we have demonstrated that Dr Warehouse may be used to predict the rate of spread of any disease for which there is an adequate compartmental model. Given the fact that alterations of the basic models are constantly created and evaluated, the extension mechanism in the system is suitable for the timely testing of a model for a new disease or variant. In the remainder of the subsection, we give an overview of the supported models.

1) Simple SIR Model

The simple SIR (Susceptible/Infected/Recovered) model derives its name from the three compartments that are used to model a population struck by a disease: susceptible (S), infectious (I), and recovered (R). A susceptible individual from the S compartment may become infectious through contact with an infectious individual from the I compartment, while an infectious individual becomes a member of the R compartment after a recovery period and develops a lasting immunity. The rate at which a disease is transmitted from an infectious to a susceptible individual is the contact rate β, while a rate at which an infectious individual recovers is the recovery rate γ. Actual values for the rates β and γ depend on the disease that is being modelled. Some of the diseases that may be described by this model are influenza, measles, and acute hepatitis C. Three ordinary differential equations describe the dynamics:

$$dS / dt = -\beta I S, \quad (1)$$
$$dI / dt = \beta I S - \gamma I, \quad (2)$$
$$dR / dt = \gamma I. \quad (3)$$

In order to analyse the dynamics of a disease within a population, a user of the Dr Warehouse system has to specify the necessary parameters: S, I, R, β and γ. As an aid in this process, the extension may automatically provide the recommended values of β and γ for a disease that has been selected by the user from the list of the disease supported by the extension. The compartment sizes (S, I, and R) may be automatically estimated from the previous disease cases once the user specifies the coefficient describing the share of the available data set with cases in the complete population, and the periods from which the number of infected and recovered individuals should be determined

In Fig. 11, we give an example of a prediction that was generated by a chronological simulation for influenza using our implementation of the simple SIR model. The presented chart demonstrates a typical situation when equilibrium in a population is gradually reached after a peak in the number of infected individuals.

2) Generalized SIR Model

The generalized SIR model is an extension of the simple SIR model, in which demographic processes are acknowledged: the appearance of new susceptible individuals through birth and the disappearance of individuals from all three compartments (S, I, and R) because of death. The inclusion of these processes in the model is especially suitable when analysing the dynamics of a disease over a longer period. In addition to the parameters of the simple SIR model, there is the mortality rate μ, whose value is often used for the birth rate as well. The equations used in this model are slightly more complex from those in the simple SIR model owing to the terms with the μ rate:

$$dS / dt = \mu - \beta I S - \mu S, \quad (4)$$
$$dI / dt = \beta I S - \gamma I - \mu I, \quad (5)$$
$$dR / dt = \gamma I - \mu R. \quad (6)$$

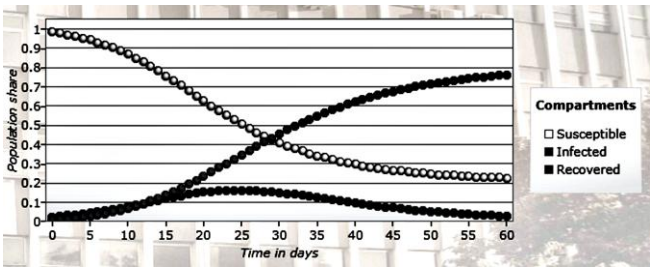


Figure 11. Example of a disease forecast obtained using the simple SIR model.

3) SIS Model

The SIS model could be considered a narrower version of the simple SIR model owing to the absence of the compartment containing recovered individuals. In this model, once an individual stops being infectious, there is no immunity period. As a result, this individual immediately becomes susceptible to the modelled disease. This scenario is typical of many sexually transmitted infections. There are two equations modelling such diseases:

$$dS / dt = \gamma I - \beta I S, \quad (7)$$
$$dI / dt = \beta I S - \gamma I, \quad (8)$$

In Fig. 12, there is an example of a prediction obtained from using the SIS model.

4) SIRS Model

The SIRS model is another variation of the SIR models, in which immunity to the modelled disease is lost after a limited period, i.e., a recovered individual eventually becomes susceptible. Because of this addition, there is a new parameter ω - the rate at which the immunity is waning. A system of equations that take into account these events and basic demographic processes include:

$$dS / dt = \mu + \omega R - \beta I S - \mu S, \quad (9)$$
$$dI / dt = \beta I S - \gamma I - \mu I, \quad (10)$$
$$dR / dt = \gamma I - \omega R - \mu R. \quad (11)$$

5) SEIR Model

The SEIR model is more complex than the aforementioned models as it features additional compartment E, which contains exposed individuals. A susceptible individual may become first exposed and only then infectious. Exposed individuals are infected but not infectious, i.e., they cannot transmit the disease for a certain period that may be expressed using the σ rate. The equations that model such dynamics include:

$$dS / dt = \mu - \beta I S - \mu S, \quad (12)$$
$$dE / dt = \beta I S - \mu E - \sigma E, \quad (13)$$
$$dI / dt = \sigma E - \gamma I - \mu I, \quad (14)$$
$$dR / dt = \gamma I - \mu R. \quad (15)$$

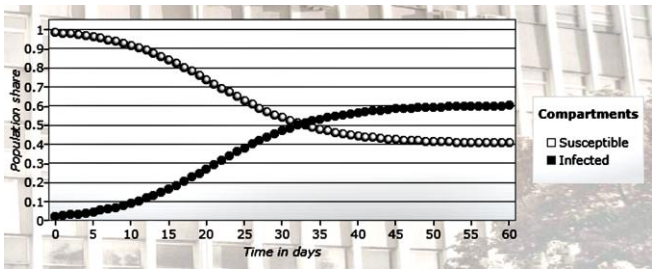


Figure 12. Example of a disease forecast obtained using the SIS model.

VI. Conclusion and Future Work

We provided an extended overview of a software system that may be used in public healthcare and epidemiology for data collection, data mining, analyses, monitoring, and research. The main contribution is the construction of a versatile and comprehensive software solution for epidemiology, as opposed to the similar existing solutions. It should act as a central point for the collection of epidemiological data, their analysis, and presentation tailored to different groups of intended users: public healthcare professionals, epidemiologists and general public. We expect that this system may have an important role in the activities concerned with epidemic control owing to the several already implemented compartmental models for the prediction of disease dynamics. The provided data visualization and reporting controls should offer better understanding of temporal and spatial disease patterns. In addition to a general description of the system's architecture, data source, and client applications, a special attention was given to the implementation of the data warehouse and data analyses. The presented examples of analyses illustrate some of the results that may be obtained through the system using the available data sample. In order to support application of the latest epidemic models and their evaluation in the context of the collected data, we incorporated an extensibility mechanism that allows addition of new functionalities to the system.

We demonstrated through various examples that the developed solution is operational and supports epidemiological monitoring, research, and prediction. These capabilities are illustrated on a data sample from an institution of public healthcare. However, the actual value brought by this solution could only be determined after its prolonged use. During that period, epidemiological data should be collected within the featured data warehouse. Since many epidemiological analyses require data sets covering large time spans, the system should be running for at least several years before it could be systematically evaluated. By creating Dr Warehouse, we have provided a software foundation for epidemiological forecasting through two predictive approaches that have been extensively and successfully used in practice: mathematical modelling of disease dynamics and data mining of epidemiological data. In addition to using available implementations of predictive procedures, epidemiologists may make customizations and even add completely new procedures to the system. This is a more probable research scenario because epidemiological characteristics of many diseases are region-dependent and may change over time, which in turn requires new ways of predicting epidemics and disease dynamics. For these reasons, the actual potential of the system may only be achieved if the system is adopted and readily used by the domain experts.

There are numerous ideas for future work and research on the presented system. We may modify the existing analyses and make them more generic so that they could support a greater number of queries. Moreover, we intend to implement and test several models that are based on cellular

automata. The data warehouse schema may be altered and extended in order to support additional analyses. Since low quality data within such solutions may cause "unnecessary anxiety, investment of time, and expensive engagements with healthcare professionals" [42], we plan a more elaborate assessment of data quality as part of future work. Furthermore, based on these assessments, we intend to improve the ETL process as it is considered a key data quality factor. Given the fact that data quality varies according to user experience, among other factors, we intend to utilize the assessment method for subjective quality properties and the Data Quality Manager (DQM) Prototype presented in [34]. The presented DQM is a prototype for the assessment of data quality within heterogeneous databases that incorporates a data quality assessment framework based on extensions of the Reference Model, Measurement Model, and Assessment Model. Moreover, it takes into account the type of information system when assessing data quality. We consider such property important because of the high data quality requirements associated with the application area.

We may also enforce a strict security policy by introducing user roles and separating the set of functionalities into subsets better suited for various user categories. A new version of the system could be implemented using open (and free) technologies, which could lead to a creation of a completely open version of the system. Due to the prominence of spatio-temporal and epidemiological data in Dr Warehouse, best practices from geographic information systems and constraint databases are topics also worth exploring in the future. Furthermore, significant additions to the system would be the construction of an epidemiological knowledge base, which could be regularly updated or consulted during data analyses, together with the inclusion of a convenient ontology. We have already undertaken some of these activities by integrating information from Freebase into Dr Warehouse. With such enhancements, the semantics may be expressed and the new version of the system could communicate with other systems that follow the idea of the Semantic Web.

ACKNOWLEDGEMENT

The research was supported by Ministry of Education and Science of Republic of Serbia, Grant III-44010. The authors are most grateful to The Institute of Public Health of Vojvodina in Novi Sad for the provided absenteeism data sample and valuable comments.

REFERENCES

[1] V. Ivančević, M. Knežević, M. Simić, I. Luković, and D. Mandić, "Dr Warehouse - An Intelligent Software System for Epidemiological Monitoring, Prediction, and Research," Proceedings of the 5th IARIA International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2013), Jan.-Feb. 2013, pp. 204-210.

[2] W. R. Dowdle, "The Principles of Disease Elimination and Eradication," Bulletin of the World Health Organization, vol. 76, Suppl 2, 1998, pp. 22-25.

[3] T. G. Savel and S. Foldy, "The Role of Public Health Informatics in Enhancing Public Health Surveillance," CDC MMWR – CDC's Vision for Public Health Surveillance in the 21st Century, vol. 61, 2012, pp. 20-24.

[4] A. B. Bernstein and M. H. Sweeney, "Public Health Surveillance Data: Legal, Policy, Ethical, Regulatory, and Practical Issues," CDC MMWR – CDC's Vision for Public Health Surveillance in the 21st Century, vol. 61, 2012, pp. 30-34.

[5] H. Rolka, D. W. Walker, R. English, M. J. Katzoff, G. Scogin, and E. Neuhaus, "Analytical Challenges for Emerging Public Health Surveillance," CDC MMWR – CDC's Vision for Public Health Surveillance in the 21st Century, vol. 61, 2012, pp. 35-39.

[6] K.M. Sullivan, A. Dean, and M.M. Soe, "OpenEpi - a web-based epidemiologic and statistical calculator for public health," Public Health Reports, vol. 124, no. 3, May-June 2009, pp. 471-474.

[7] "Open Source Epidemiologic Statistics for Public Health," http://www.openepi.com/ [Dec. 15, 2013].

[8] J.H. Abramson, "WINPEPI updated: computer programs for epidemiologists, and their teaching potential," Epidemiologic Perspectives & Innovations, vol. 8, no. 1, February 2011, pp. 1-9.

[9] "Epi Info™ - Community Edition," http://epiinfo.codeplex.com/ [Dec. 15, 2013].

[10] "HealthMap," http://www.healthmap.org/ [Dec. 15, 2013].

[11] "Outbreak Watch Social Biosurveillance Network," http://www.outbreakwatch.com/ [Dec. 15, 2013].

[12] "Google Flu Trends," http://www.google.org/flutrends/ [Dec. 15, 2013].

[13] Strategija razvoja informacionog društva u Republici Srbiji do 2020. godine [The Strategy for the Development of Information Society in the Republic of Serbia until the Year 2020], (in Serbian), Službeni glasnik Republike Srbije, vol. 51, 2010.

[14] M. Fimić, M. Radulović, I. Vulić, and S. Atanasijević, "Zdravstveni informacioni sistem Ministarstva odbrane Republike Srbije – generičko rešenje za integraciju institucija" [The Health Information System of the Ministry of Defense of the Republic of Serbia – A Generic Solution for Institution Integration], (in Serbian), Proceedings of YU INFO 2012, pp. 511-516.

[15] G. Johns, "absenteeism," in The Blackwell Encyclopedia of Sociology, G. Ritzer, Ed. Oxford, UK: Blackwell Publishing, 2007, pp. 4-7.

[16] "Absence management, Annual Survey Report 2009 - CIPD," http://www.cipd.co.uk/NR/rdonlyres/45894199-81E7-4FDF-9E16-2C7339A4AAAA/0/4926AbsenceSRWEB.pdf [Dec. 15, 2013].

[17] Đ. Jakovljević and P. Mićović, Zdravstveno stanje i zdravstvene potrebe stanovništva Srbije [Health Status and Health Needs of the Population of Serbia], (in Serbian), http://www.palgo.org/files/leaflet/brosura_zdravstvo.pdf [Dec. 15, 2013].

[18] "Dr Warehouse," http://www.acs.uns.ac.rs/sr/node/237/1429892 [Dec. 15, 2013].

[19] "SQL Server Analysis Services," http://technet.microsoft.com/en-us/sqlserver/cc510300.aspx [Dec. 15, 2013].

[20] "Microsoft SQL Server," http://www.microsoft.com/sqlserver/ [Dec. 15, 2013].

[21] "Microsoft Integration Services," http://msdn.microsoft.com/en-us/library/ms141026%28v=sql.105%29.aspx [Dec. 15, 2013].

[22] "SQL Server Agent," http://technet.microsoft.com/en-us/library/ms189089.aspx [Dec. 15, 2013].

[23] "International Classification of Diseases," http://www.cdc.gov/nchs/icd/icd10cm.htm [Dec. 15, 2013].

[24] J. Mundy, W. Thornthwaite, and R. Kimball, The Microsoft Data Warehouse Toolkit: With SQL Server 2008 R2 and the Microsoft Business Intelligence Toolset, 2nd ed., Indianapolis, IN: Wiley Publishing, Inc., 2011.

[25] "Microsoft Silverlight," http://www.microsoft.com/silverlight/ [Dec. 15, 2013].

[26] "Microsoft WCF Services," http://msdn.microsoft.com/en-us/library/dd456779.aspx [Dec. 15, 2013].

[27] "SQL Server Reporting Services," http://msdn.microsoft.com/en-us/data/ff660783.aspx [Dec. 15, 2013].

[28] "Celik API," http://ca.mup.gov.rs/Celik%20api%20Windows%20v1.1.pdf [Dec. 15, 2013].

[29] "PivotViewer," http://www.microsoft.com/silverlight/pivotviewer/ [Dec. 15, 2013].

[30] M. Knežević, V. Ivančević, and I. Luković, "A context-sensitive support system for medical diagnosis discovery based on symptom matching," Proceedings of the 5th KES International Conference on Intelligent Decision Technologies (IDT 2013), vol. 255, Amsterdam: IOS Press, Jun. 2013, pp. 1-10.

[31] "Microsoft Windows Phone," http://www.microsoft.com/windowsphone/ [Dec. 15, 2013].

[32] "TerraService.NET: An Introduction to Web Services," http://research.microsoft.com/apps/pubs/?id=64154 [Dec. 15, 2013].

[33] "Managed Extensibility Framework," http://msdn.microsoft.com/en-us/library/dd460648.aspx [Dec. 15, 2013].

[34] M. d. P. Angeles and F. J. García-Ugalde, "Subjective Assessment of Data Quality considering their Interdependencies and Relevance according to the Type of Information Systems," International Journal On Advances in Software, vol. 5, 2012, pp. 389-400.

[35] R. S. Sandhu, "Role-based Access Control," Advances in Computers, vol. 46, 1998, pp. 237-286.

[36] "Freebase," http://www.freebase.com/ [Dec. 15, 2013].

[37] "JavaScript Object Notation," http://www.json.org/ [Dec. 15, 2013].

[38] W.O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," Proceedings of the Royal Society of London, vol. 115, no. 772, pp. 700-721, August 1927.

[39] R.M. Anderson and R.M. May, "Population biology of infectious diseases: Part I," Nature, vol. 280, no. 5721, August 1979, pp. 361–367.

[40] R.M. May and R.M. Anderson, "Population biology of infectious diseases: Part II," Nature, vol. 280, no. 5722, August 1979, pp. 455–461.

[41] M.J. Keeling and P. Rohani, Modeling Infectious Diseases in Humans and Animals, Princeton, NJ: Princeton University Press, 2007.

[42] R.W. White and E. Horvitz, "Cyberchondria: Studies of the escalation of medical concerns in Web search," ACM Transactions on Information Systems, vol. 27, no. 4, pp. 23:1–23:37, 2009.

Hierarchical Quarters Model Approach toward 3D Raster Based Generalization of Urban Environments

Alexey Noskov
Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology
Haifa, Israel
e-mail: noskov@technion.ac.il

Yerach Doytsher
Mapping and Geo-Information Engineering
Technion – Israel Institute of Technology
Haifa, Israel
e-mail: doytsher@technion.ac.il

Abstract—**The suggested method for 3D generalization of groups of buildings in urban environments is based on the rasterization of the 2D footprints of 3D buildings. The rasterization is processed within quarters, which are automatically defined by using Digital Elevation Model (DEM), water objects and roads. Quarters were organized into a hierarchical model according to the gaps between the quarters and the stages of the clustering process. Each degree of generalization corresponds to some level of hierarchy. The 3D urban perspective is computed based on separate levels of generalization of each quarter as a function of its distance from a pre-defined view point. The developed approach enables to compile a 3D scene of urban environment based on the generalized buildings' layer. The buildings' layer consists of objects with different degree of generalization level which is growing gradually from the view point. The two main distinctions of the approach from others are: (1) the generalization is implemented with respect to the geospatial properties of urban environments and the relations between the objects; (2) the approach is simple and universal which enables to simplify the whole area of a city and can be applied to different types of cities.**

Keywords-Generalization; 3D urban model; groups of buildings; hierarchy of quarters

I. INTRODUCTION

3D generalization of the urban model is a fast-growing topic. The main types of objects in the 3D city model are buildings. Nowadays, 3D models are used in many disciplines [33]: GPS navigation, desktop and mobile city viewers, geo-simulation, architecture, and many others. The two common problems which usually arise in any discipline are: (1) huge computer resources are required for drawing 3D models based on the original, non-simplified buildings, and (2) 3D models based on the original non-simplified buildings are very detailed and often appear unreadable and overly complex. To resolve both problems we have to generalize the buildings. There are two different tasks in the building generalization process: (1) simplification of a single building, and (2) generalization of groups of buildings. The topic "simplification of a single building" is a widely researched topic; we can describe several different approaches of generalization, all of them valid. In contrast, "generalization of a group of buildings" has only been treated, so far, on a very limited level. There are several very similar approaches, largely based on the Delaunay Triangulation (DT) (e.g., [40] and [25]). We propose, in concept, another approach for the generalization of groups of buildings, based on rasterization and vectorization operations, which are carried out by sub-dividing the urban neighborhood into quarters. This paper is structured as follows: the related work is considered in section two, the source data are described in section three, the algorithms of raw quarter calculations and building quarters' hierarchy are presented in sections four and five, the raster based algorithms of generalizing group of buildings is considered in section six, the results are evaluated in section seven and, finally, in the last section the conclusions are detailed.

II. RELATED WORK

One of the most holistic approaches to the 3D generalization of buildings was described in [40]. The main idea supposes that, within a threshold (distance from a view point), we will generate objects which contain the results of simplification of single buildings, whereas outside of the threshold we will generate objects containing the results of groupings of buildings and their simplification as a single building. An approach of "converting 3D generalization tasks into 2D issues via buildings footprints" was described by He et al. in [11].

The generalization of 3D building data approach [7], based on scale-space theory from image analysis, allows the simplification of all orthogonal building structures in one single process. Another approach [36] considers buildings in terms of Constructive Solid Geometry (CSG). In [40] an approach was proposed which realized 3D single building simplification in 5 consecutive steps: building footprint correction, special structure removal, roof simplification, oblique facade rectification and facade shifting. A very interesting approach was proposed by Kada in [16] and [17]. In this approach, geometric simplification was realized by remodeling the object by means of a process similar to half-space modeling. Approximating planes are determined from the polygonal faces of the original model, which are then used as space dividing primitives to create facade and roof structures of simpler shapes.

The second aspect of 3D generalization of an urban environment is the generalization of groups of buildings. 3D generalization of groups of buildings is mentioned in several publications (e.g. [8], [10], [11], [37]). These papers describe different approaches to 3D grouping and group generalization: grouping of building models (using the

infrastructure network) and replacing them with cell blocks, while preserving local landmarks [8]; "express different aspects of the aggregation of building models in the form of Mixed Integer Programming problems" [10]; and, grouping of building models "with a minor height difference and the other with a major height difference" [11].

2D building generalization algorithms should also be considered for use by researchers for a 3D building group generalization. A holistic and automated generalization method based on a pseudo-physical model was considered in [15]. An approach based on Delaunay triangulation, Graph and Gestalt theory was described by Li et al. in [25].

In the above-mentioned publications, different approaches were considered, but we can identify some common ideas which are important for most research in this area.

In most cases it is very useful to generate levels-of-detail (LOD); normally, researchers use 3 or 4 LODs ([4], [27] and [36]). LODs are widely used in 3D video games, usually for detailed objects; more simplified objects are created for saving processor load and virtual memory [27]. Usually a detailed object has references to several simplified versions (at different levels of simplification), so that if the object stays near the view point, the most detailed version of the object is used, and as the object is located further from the view point the more simplified object is used.

It is very popular to use CityGML standard for 3D urban models ([9], [13], [19], [20], [21], [22], [23] and [35]). This format supports many useful possibilities, which are very important for working with 3D urban models (e.g., LODs, topology, semantics etc.).

Today, 3D city visualization is an extremely fast-developing topic [3]. There are many benefits to using 3D city models, for instance, the significant advantages of using 3D maps in cadastral systems and public participation in urban planning processes have been described by Shojaei et al. in [34] and by Wu et al. in [39]. There are many approaches and technical solutions for storing and visualizing 3D city models. In [31] a detailed analysis of existing approaches to 3D city visualization was published. According to this publication, there are several principal formats and standards which are normally used in reviewed projects - namely - CityGML, KML/CALLADA, X3D, X3DOM, HTML5/WebGL, OpenStreet map data format. Several data sets store huge amounts of 3D buildings' models - Paris, Berlin, Mainz, Blacksburg geodatabases and OpenStreetMap data were described. In addition, important applications for working with 3D virtual city data - CityServer3D, 3DCityDB, IGG Web 3D Service, OSM-3D Web 3D Service, HPI 3D Server and Web View Service, Xnavigator, InstantReality Player, BSContact Geo,HPI 3D WVS Clients, Google Earth - were reviewed.

The OpenStreetMap crowdsourcing project has made it possible to depict 3D maps of numerous cities world-wide. For a huge number of building models, users are defining tags (attributes) such as building heights, number of floors, type of roofs etc. The OpenBuildingModels platform [38] enables us to prepare and add to the OpenStreetMap realistic 3D complex buildings models. This, and the fact that the OpenStreetMap is a free open source project and allows access to the data under a copy-left license, enables the development of applications for creating 3D interactive real maps. Impressive results were achieved in the OSM-3D project (see [29]). This is an experimental but actual project, working in 3D globe application, and has been released as Java Applet.

As mentioned by Hildebrandt and Döllner in [12], due to the advances in computer graphics and improved network speed it is now possible to navigate in a 3D virtual world in real time. Until recently, the technologies employed required installing standalone applications or plugins on navigators. The relatively new HTML 5 format brings new solutions for visualizing 3D data in a web browser by using WebGL. Several globe projects have proven that such technologies are feasible and can be employed. One of these projects was described by Mao and Ban in [28], where CityGML data are interactively converted to X3D format according to the user request on the server, and the X3D data are visualized on the user's web browser. This method can work on any modern web browser with WebGL (e.g., Mozilla Firefox, Google Chrome, and Safari). In [30] it has been proven that the same approach can effectively work on mobile devices (smart-phones and tablets).

In spite of the large number of publications and developing projects, we can identify a very important shortcoming on almost all 3D city maps (or screenshots). There are usually only two ways of displaying large cities: depicting only the buildings nearest to the view point (whereas all the other buildings are not displayed and the area is depicted as a plain map), or displaying all the 3D building models (which usually causes a long processing time and heavy computer and internet traffic resources, and furthermore, causes distant parts of the city to be presented as a very dense and unreadable 3D view). As mentioned above, we are seeing a large number of publications on generating and using LODs of single buildings, while there are only a very limited number of approaches to creating LODs of group of buildings in order to solve the problem described above.

Additionally, it must be mentioned that in the approach we used Kohonen's self-organizing maps [18] (one of the artificial neural network algorithms) to classify quarters according to a set of different attributes.

There are several possible approaches to multidimensional classification. In our case, using one of the clustering algorithms seems very promising. There are several common groups of clustering algorithms: hierarchical clustering, centroid-based clustering, distribution-based clustering, and density-based clustering. According to [14] "hierarchical clustering is based on the core idea of objects being more related to nearby objects than to objects farther away". "As such, these algorithms connect 'objects' to form 'clusters' based on their distance". This method is often considered obsolete. Centroid-based clustering is based on defining the optimal central vector of clusters. K-means or Lloyd's algorithm [26] is a well-known centroid-based approach. It is an unsupervised algorithm which requires setting the number of classes. It has been

mentioned that a k-means approach cannot find non-convex clusters [6]. Some k-means algorithms to classify the raw quarters of Trento have been tested, and the results look interesting and useful for our aims. According to [32] distribution-based clustering methods "suffer from one key problem known as over-fitting, unless constraints are put on the model complexity". In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set [24]. The DBSCAN [5] density-based popular clustering method has serious disadvantages for our case, as it is not easy to define initial parameters (certain distance thresholds and minimum number of points required to form a cluster) for this method. This disadvantage was partially eliminated in the OPTICS [2] method; in this method only a minimum number of points are required. But it is still a problem, because in our case we have a very specific small group of quarters. On the other hand, if we set a small minimum number of points we will get too many clusters. From all the approaches described, the k-means algorithm seems very suitable for our investigation. The Kohonen's self-organizing map (SOM) [18] is one of the ANN methods; this method's implementation is very close to k-means. SOM is not only a clustering method; it is also a very useful tool for visualization and evaluation of the results of clustering. At this time we are focusing our attention on the SOM approach of clustering.

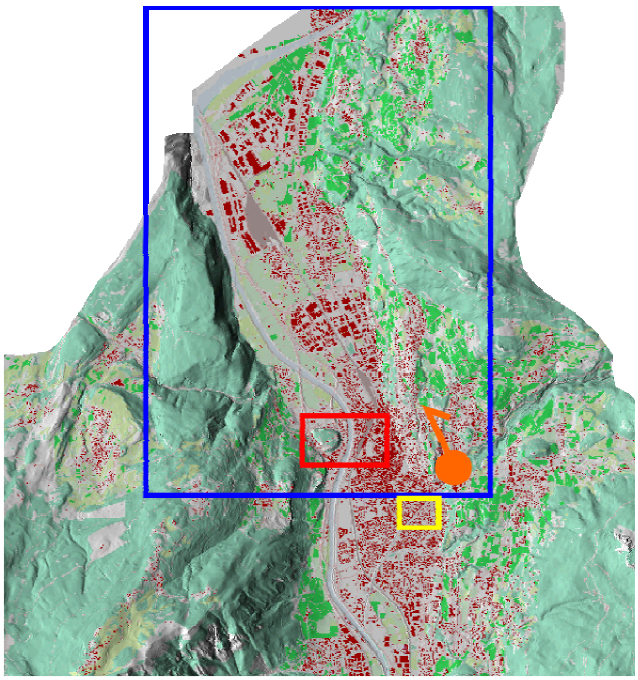


Figure 1. Map of Trento.

III. SOURCE DATA

For implementing and testing our approach, the free geodata of the city of Trento, Italy was used. The buildings (with individual heights), water objects and roads were extracted from the landuse map of Trento (see Fig. 2). The landuse map and land relief (DEM) were downloaded from the website of Trento Municipality [41]. On the map of Trento (see Fig. 1) the buildings are depicted as brown areas; the extent of the maps in Fig. 2, Fig. 3 and Fig. 4 are marked with a red square; the extent of the map on Fig. 13 with a blue square, and the view point and view direction of Fig. 14 with an orange circle and arrow.

IV. CALCULATION OF RAW QUARTERS

Finding a realistic method of simplification is a very important issue in generalization. One of the more common problems is when buildings are joined through obstacles such as wide roads or rivers. In this case, buildings must not be joined to each other, and these buildings from the two sides of the obstacle should be merged with other, more distant objects, which are, however, located on the correct side of the obstacle. To resolve this problem, we decided to split the urban space into quarters which are divided by the main, significant objects. These objects cannot be involved in the generalization itself.

To calculate the quarters, we decided to use the slope of the terrain, water objects and roads. In Trento, it was found that buildings are positioned only on areas with a slope of less than 30 degrees. Accordingly, areas with slopes greater than 30 degrees of the terrain were excluded. We also used roads and water objects for defining the quarters. These objects are polygons extracted from the landuse map. In an initial work [1] we used attributed data of roads, based on the fact that we worked with linear objects downloaded from the OpenStreetMap website. In the next stage, presented in this paper, we use polygonal features, which, due to their geometric characteristics (their size) as the main parameter, avoid the need to use attributed data (like type of object) of roads and the water objects. All these three classes – slopes, roads and water objects - were merged into one raster map with 1 meter resolution (which has been found to be adequate for small-scale urban generalization).

The raster map of the merged objects is the base for quarter calculating; further processing can be divided into several consequent steps.

The raster transformations for splitting the city into quarters have been selected because the standard vector approaches (e.g. polygons based on vector roads) have several limitations. The source vector road data may contain features such as unfinished roads, dead end roads, etc., features affecting its topological correctness. Splitting the area into quarters based on these data might result in a very complex polygonal map containing artifacts. In contrast, the raster transformations approach enables to exclude most of the artifacts and the unnecessary boundaries and vertices. As the width of the narrowest roads is about 2-3 meters, the resolution of the raster maps has been defined as 1 meter. Accordingly, the quarter map is composed of polygon bounds which coincide approximately with the road centerlines (±1 meter), as well as not intersecting the buildings.

A. *Region growing of base features*

All pixels of the merged objects got the value "1"; empty space on the raster map got the "Nil" value. Each group of

pixels with value “1” has been expanded by adding one pixel (1 meter) and the results are depicted in Fig. 3.

B. *Inverting of pixel map*

At this step, the values of pixels were inverted (“1” to “Nil” and vice versa), resulting in many pixel areas with the value “1” which are split by “Nil” pixels.

C. *Defining quarter areas having unique values*

To set a unique pixel value to each quarter area, we vectorized the raster map. Each vector that defines a polygonal object got a unique integer identifier. Polygons not containing buildings were removed. Then the vector map was rasterized. For raster values, polygon identifiers were used. As a result (see Fig. 4), we got a raster map with groups of pixels (a "quarter") and each group (which is separated by “Nil” pixels from the adjacent group) got a unique integer identifier.

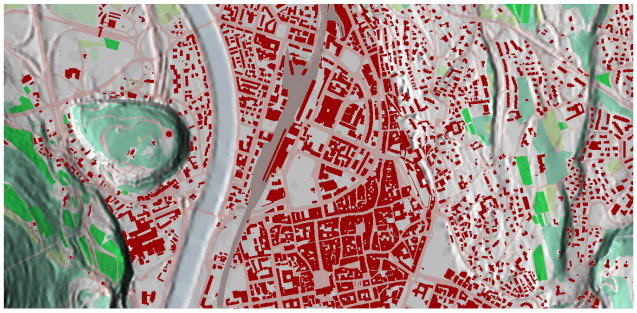


Figure 2. Shaded Landuse Map of Trento and Buildings (brown polygons).

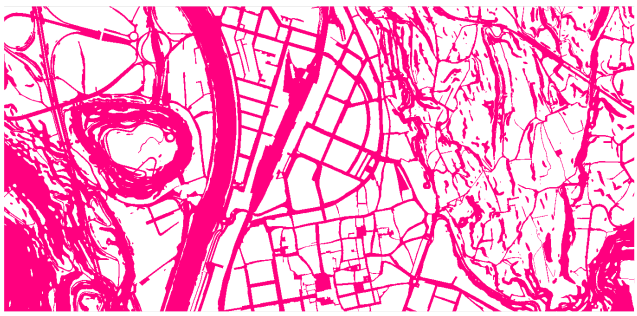


Figure 3. Non-Nil Pixel Groups which Split the City Space into Quarters.

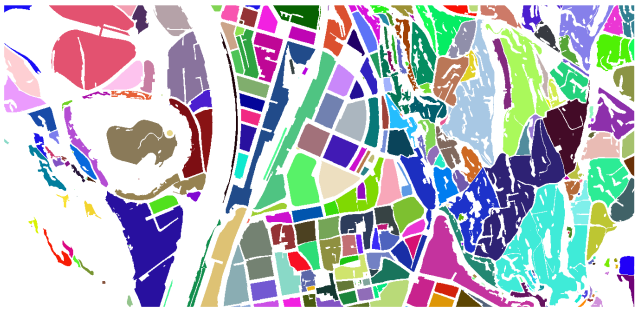


Figure 4. Inverted Raster Map with Unique Pixel Values.

At this stage, a set of raw quarters was prepared. The next data processing was based on this. We can see on the map that quarters contain holes, dead-ends and unfinished roads. These elements naturally disappear during the generalization of the quarters.

V. CALCULATION OF QUARTERS’ HIERARCHY

In the previous step we prepared raw quarters. We cannot use them for a high level of generalization, as raw quarters would be too small for large generalized buildings. To overcome these limitations, a new flexible hierarchical approach of subdividing the urban area into variable quarters was developed. The raw quarters are placed on the lowest level of the hierarchy; on the highest level, the whole area of the city is defined as one large quarter. A special approach to developing the quarter generalization (or quarters merging) will enable this hierarchy, where the size and content of the quarters will be correlated with the level of the 3D generalization, and this, in turn, will be related to the distances from the view point. Each level of the hierarchical tree of quarters has some level of quarters’ generalization. The hierarchy is based on buffer operations. We will widen the quarters by a buffer; thus, adjacent quarters will be merged into one object, while other objects will only change their geometry (outer boundary of quarters will be simplified, some inner small elements like holes and dead-end roads will be filled). Then we will decrease the quarter by a buffer with the same size. If the width of the gap of merged quarters is smaller than the buffer width, the objects remain as a merged polygon, otherwise the polygon will be split back into separate objects differing from the original objects due to their simplified geometry.

The raw quarters will be generalized and organized in a hierarchical tree. Each level of tree is built on the previous level, and is calculated as follows: first, the attributes of each quarter are calculated, depending on their classification. Each level is built according to some base buffer width. All quarters are separated into temporary layers according to their classes. Then, we apply buffer operations (with a width which is equal to two times the base buffer width) separately to each layer, and overlay all the layers into one map. On this map, we apply buffer operations with the base buffer size to all quarters without taking their classes into consideration. This suggested approach helps to merge quarters of the same class faster than others (i.e., quarters of different classes should be at least twice as close as quarters of the same class to be merged). In Listing 1 the algorithm of this process is presented. In Fig. 5 you can see its results (A: Background Color by Original Classified Quarters, Black Polylines are Outlines of Buffers by Classes of Quarters, Width is Equal to Twice the Base Width; B: Withdrawal of Buffers; C: Adding Base Buffer Width to the Polygons; D: Withdrawal of Buffers –Resulting as the Final Buffering).

The first step of quarter generalization is calculating the attributes for each quarter. These calculated attributes (a list of the attributes is depicted in Fig. 6) will be used for the classification of the quarters.

We decided to use the Kohonen's Self-Organizing Map approach to classify quarters. The number of clusters was defined for all levels in the hierarchy of the quarters to perform classification. There are several techniques to

Listing 1. Algorithm of Building the Hierarchical Tree of Quarters.

```
quarters=original_quarters
for ( buffer=1 , buffer+=0.2 , buffer < 2):
    Calculate attributes of quarters
    Classify quarters according attributes
    for ( i=0 , i++, i < number of classes):
        Extracting class i into separated map
        foreach current_quarter in quarters:
            Adding buffer*2 to current_quarter
            Withdrawing of buffer*2 from current_quarter
        Merging quarters from separated maps into
        buffered_quarters
        foreach current_quarter in buffered_quarters:
            Adding buffer to current_quarter
            Withdrawing of buffer from current_quarter
    Appending of buffered_quarters into result Hierarchical Tree
    quarters=buffered_quarters
```

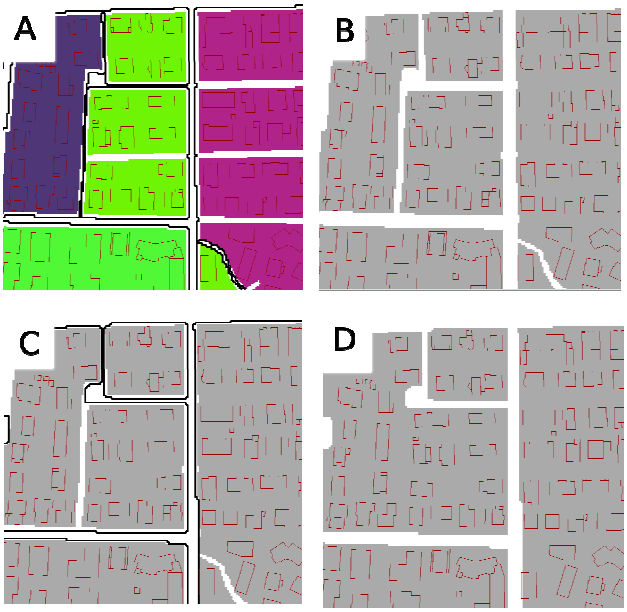


Figure 5. Buffering Process.

automatically define the numbers of clusters (e.g., a gap statistic approach, an information theoretic approach, etc.). At this time, we have focused on a manual definition of the number of classes. An initial manual analysis of a series of maps based on quarter attributes visualization was carried out. On Fig. 6 you can see example of quarters' classification, meaning of parameters are presented below:

- A - Typical Azimuth of Buildings' Sides;
- S_q – Aarea of Quarter;
- S_b – Area of Buildings in a Quarter;
- D – Density of Buildings, S_b/S_q;
- F_q - Fractal Dimension of the Quarter's Boundary, 2·log(perimeter) / log(area);
- F_b – Mean Fractal Dimension of Buildings' Boundaries in a Quarter;
- C_q – Compactness of a Quarter, perimeter/(2· √(π·area));
- C_b – Mean Compactness of Buildings;
- P_q – Perimeter of a Quarter;
- P_b – Mean Perimeter of Buildings;
- PP – P_b/P_q;
- H - Arithmetic Weighted Mean by Areas of the Heights of Buildings.

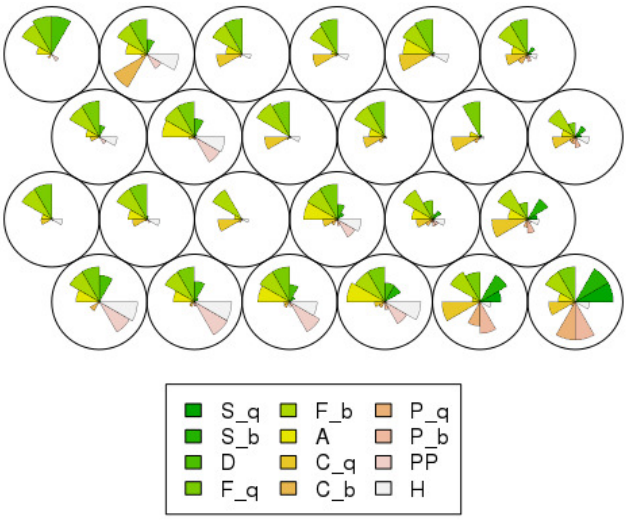


Figure 6. Diagram of Codebook Vectors (Centers of Clusters). 24 Classes of Calculated Quarters with Buffer Width of 1.6 meter..

As aforementioned, in order to take into account the quarters' classes, adding a weighted buffer to the quarters has been suggested (thus differentiating between quarters of the same class and quarters of different classes), aiming to merge neighboring quarters. To avoid vector artifact and topology problems, data was converted to raster, and buffering operations were executed in a raster environment. We used the raster resolution as the base width of the buffer. Not only does the buffering phase provide the possibility of merging quarters, but this operation also helps to fill holes and dead-end roads in polygons, and to eliminate small elements of quarters' boundaries. It should be mentioned that converting vector to raster can also work like a generalization operation. Generally speaking, this phase in the research, which is based on vector to raster and raster to vector operations, as well as region growing and buffer implementations on the one hand, and on the quarters' attributes on the other hand, enables us to generalize the quarters and lets us move up or down in the hierarchical level of the quarters' subdivision.

As mentioned above, quarters of the same class were merged faster than quarters of different classes. This was achieved by putting quarters of the same class into isolated

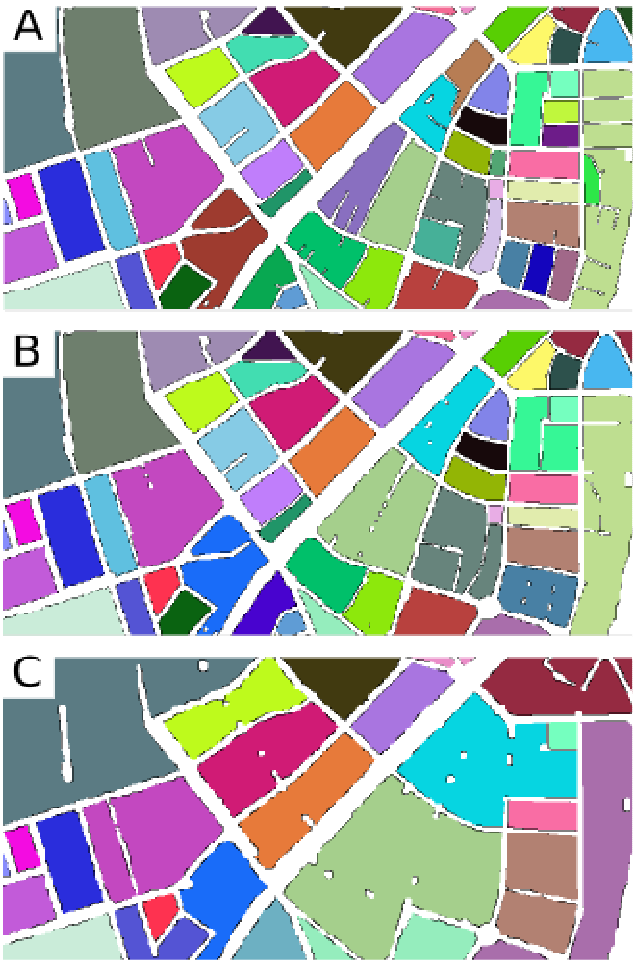


Figure 7. Quarter Buffering Generalization (Buffer Width, in meters): A: Original Raw Quarters; B - 1.4, and C - 1.8.

sub-environments (temporal layers) and using different widths of buffers (see Fig. 6 and Fig. 7).

This suggested approach allows the performing of quarter generalization based on buffering operations while taking into account quarters' classes. Using this method builds a hierarchical tree of quarters (in the current sample of Trento, from the raw source of 2679 small quarters up to a single huge quarter). We performed the generalization of quarters starting from a buffer width of 1 meter and increasing it by increments of 0.2 meter, until the buffer width reaches 2.6 meter. It was decided to start quarter generalization from 1 meter because this is the resolution which is used to generate the raw quarters; and because the upper limit of 2.6 meter as a higher buffer width generates oversized quarters.

As in the previous research [1], we decided to use 8 degrees of generalization of buildings based on rasterization processes with resolutions 10, 15, 20, 25, 30, 40, 50 and 60 meters (resolutions which correspond to degree of generalization). A graph of the varying number of quarters and the size of maximal quarter (see Fig. 8) was used to define which levels of hierarchy can be used for further processing. In addition, the original vector map of buildings was converted to raster maps with different resolutions (10, 15, 20, 25, 30, 40, 50 and 60 meters). These raster maps, overlaid with the generalized quarter maps, were used to estimate which resolution of buildings generalization should be used with each generalized quarter map. Overlaying of the generalized quarter map with different resolution raster maps of buildings is illustrated in Fig. 9. We can then see very clearly on the right side of the figure that the sizes of the generalized buildings are too large to use this quarter map for generalization at this resolution.

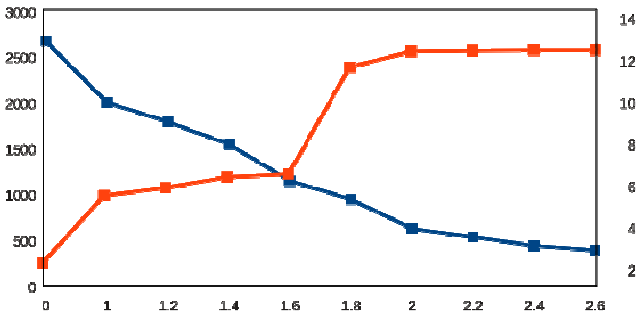


Figure 8. Graph of Number of Quarters (blue Line, left Y-axis); and the Size of Maximal Quarter (red line, right Y-axis in million sq. meters); X-axis – Base Buffer Width (in meters).

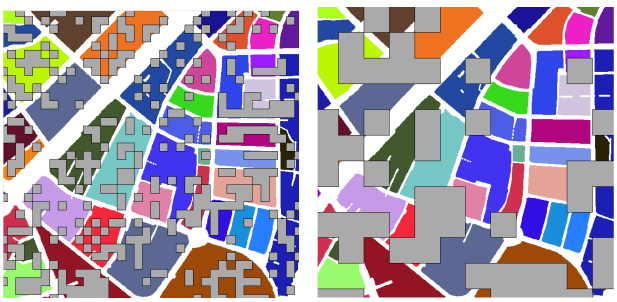


Figure 9. Quarters and Sizes of Buildings Generalization: Appropriate (left) and Too Large (right).

By using this method we estimated which levels of quarter hierarchy can be used and which resolution of buildings generalization should be processed with these levels (see Table 1).

VI. GENERALIZATION OF BUILDINGS

The fact that in urban areas, most (if not all) of the buildings have orthogonal sides, is the background for our raster-based generalization approach. Usually, in adjacent areas (quarters in our case), buildings would be spatially oriented in the same direction. Therefore, the generalization process consists of defining the typical azimuth of buildings' sides for each quarter. Once a typical azimuth is known, by applying the rasterization process in this direction, the staircase-type appearance of lines, or legs of closed polygons, which is very common in the rasterization processes, can be eliminated. A non-rotated rasterization (parallel to the grid axes) while the buildings are positioned in another orientation will result in a staircase-type appearance of the bordering lines of the buildings and too

many unnecessary vertices, which will prevent us from achieving a smooth geometry of the generalized objects.

A. *Defining the azimuth of buildings' sides*

As aforementioned, in urban areas, most of the buildings have orthogonal sides; thus, it is possible to define the average spatial orientation of the buildings. Within each quarter, the azimuths of all the buildings' sides were computed. For each building in the quarter, the longest side and its azimuth were identified. Then all the azimuths of the other sides were rotated by 90 degrees (clockwise) again and again; and the rotated azimuths (and their lengths) were put into one list. The list was sorted by lengths, and then lengths with the same azimuths (up to a predefined threshold) were averaged. A threshold of 1 degree when looking for close buildings' side azimuths has been found to give satisfactory results. A weighted average of the azimuths of the longest lengths of all the buildings within a quarter is used to define the general orientation of all the buildings of the quarter.

TABLE I. BASE BUFFER WIDTHS AND APPROPRIATE RESOLUTIONS OF BUILDINGS GENERALIZATION

Base buffer width (in meters) used to generate quarters' map (number of level in hierarchical tree)	**Resolutions of generalization, (in meters)**
original buildings (0)	original buildings
original buildings (0)	10
original buildings (0)	15
1.0 (1)	20
1.0 (1)	25
1.2 (2)	30
1.4 (3)	40
1.6 (4)	50
1.8 (5)	60

B. *Rotation and rasterization of the buildings in a quarter*

As mentioned above, and in order to significantly reduce the number of vertices of the generalized building and achieve a more realistic appearance of these simplified objects, rasterization should be carried out in the spatial orientation of the buildings. A rasterization which is spatially oriented parallel to the grid axes will define the buildings which are not oriented parallel to the grid axes in a staircase-type appearance of the buildings' sides. Accordingly, all the buildings within a quarter were rotated counter-clockwise at the angle of the general orientation of all the buildings of the quarter. Then the rotated buildings were rasterized using a certain pixel size resolution (as explained in the next section). Each pixel with more than half its area covered by the original buildings gets the value "1"; otherwise it gets the value "Nil". Fig. 10 shows the result of this stage.

The level of the generalization is a function of the pixel size rasterization process - the greater the pixel size, the greater the degree of generalization. Accordingly, each quarter has been generalized at several levels of rasterization, resulting in several layers of different levels (level-of-detail) of generalized buildings for each quarter. Based on the original data of Trento, and according to our analyses, we found that using pixel size resolutions of 10, 15, 20, 25, 30, 40, 50 and 60 meters produces satisfactory results of a continuous and consecutive appearance of the level-of-detail of the generalized buildings. Buildings were generalized independently for all quarters and at all resolutions according to Table I (each resolution corresponds to a definite level of quarter hierarchy). Generalized buildings are stored in separated layers; the identifiers of these layers contain resolution of the buildings generalization and the number of the level in the quarters' hierarchy (or actually, the buffer width).

Figure 10. The Generalization Process of Buildings in a Quarter: Original Buildings (left); Rotated Quarter and the Generalized 10 meter Rasterized Buildings in red (middle); Final Result (right).

Listing 2. Algorithm of Arranging the Quarters' List (Based on the Hierarchical Tree) to Compile a 3D Scene.

```
array=[   [ zone from view point , hierarchy level ],
          [        0-1000 m,              0        ],
                    …
          [        >8000 m,               5        ]  ]
result_list=[]
foreach current_zone, current_level in array:
    Intersect map of current_zone with current_level of quarters' hierarchy, getting current_quarters list
    foreach quarter in current_quarters:
        if (child of quarter in result_list):
            Append others child quarters of quarter to result_list
        else:
            Append quarter  to  result_list
```

To draw a 3D perspective of the city with the generalized buildings, the position of a view point had to be defined. Then we built buffer zones around the view point. The buffer zones defined the distances (practically, range of distances) from the view point to each quarter. As mentioned above, generalized buildings are grouped separately and stored by quarters in layers with identifiers containing the resolution of the buildings generalization and the level in the quarters' hierarchy. To define what layers of generalized buildings will be used in the 3D scene, we intersect the first zone (0-1000 meter from view point) with the 0-level map from the quarters' hierarchy. Selected quarters are stored in an

accumulated list containing IDs of quarters and the level of each quarter in the hierarchy. Then we take the zone next further from the view point and check on Table I what level of quarters should be used, and intersect this zone with the defined quarter layer. After that we check all the selected quarters. If a member-quarter of the selected quarter (i.e., it is part of the quarters in the lower hierarchy that compose the selected quarter in the current hierarchy) already exists in the accumulated list, we add to the list only the other member-quarters of the current selected quarter. Only quarters which do not contain member-elements in the accumulated list are added to the list. To compile a final 3D scene we just need to merge layers of the original and the generalized buildings, according to the accumulated list, into one layer. The process is repeated until the last zone is achieved. On Listing 2 a pseudocode of this described process is presented.

Fig. 11 depicts the degree of generalization for each quarter, where the colors indicate the degree of the generalization. The relationship between the distances from view point, pixel size generalization, and the colors, are described in Table II. Finally, we merged all the separate generalized layers of all the quarters into one map (see Fig. 13) for further 3D visualization. The division of distances from the view point into a scale of continuous intervals was based on several tests, which enabled us to draw a realistic and continual 3D model or perspectives. The results of a 3D visualization, and comparison of the 3D perspectives with the original buildings and with the generalized buildings, are presented in Fig. 14.

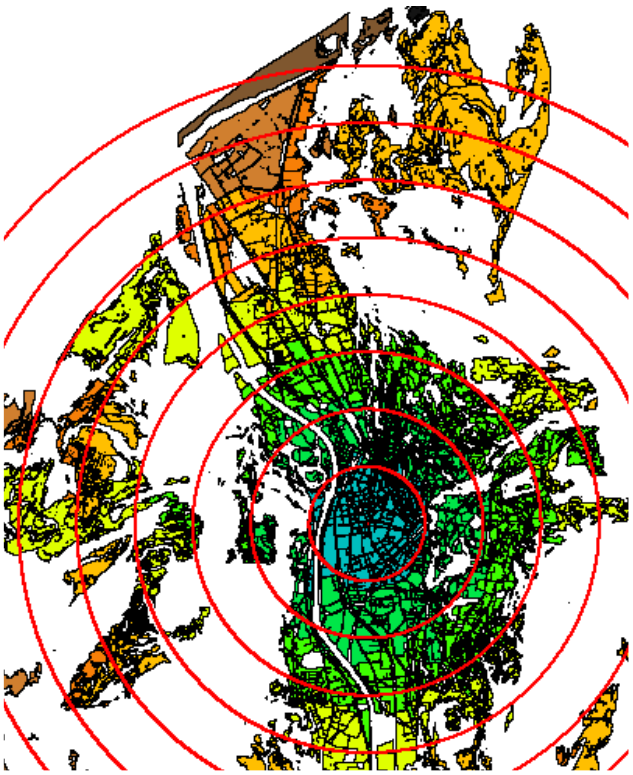


Figure 11. Defining the Degree of Generalization using Buffer Zones: Borders of Buffer Zones (red circles) and Quarter Borders (black).

TABLE II. DISTANCES FROM THE VIEW POINT, RESOLUTIONS OF GENERALIZATION, AND COLORS

Distances from view point, meters	Resolutions of generalization, meters	Background colors of the map in "Figure 12"
0 - 1000	original buildings	
1000 - 2000	10	
2000 - 3000	15	
3000 - 4000	20	
4000 - 5000	25	
5000 - 6000	30	
6000 - 7000	40	
7000 - 8000	50	
>8000	60	

VII. NUMERICAL EVALUATION

Table III presents the number of geometry primitives and the speed of the visualization process as a comparison between the original data and generalized data. As we can see, there is a significant reduction in visualization speed and in the number of polygons and nodes.

TABLE III. RESULTS OF THE GENERALIZATION

Parameter	Original building layer	Generalized building layer used for 3D visualization
Number of nodes	114,648	34,391
Number of polygons	46,339	14,956
Speed of 3D visualization, second	6.6	1.2

To evaluate the quality of generalization, the mean coefficient of building compactness was calculated for each resolution of generalization (see Fig. 12). The coefficient of compactness of a single building is equal to $\alpha=P^2/(4*\pi*A)$, where P – perimeter, A – area ($\alpha=1$ for a circle, $\alpha=1.27$ for a square).

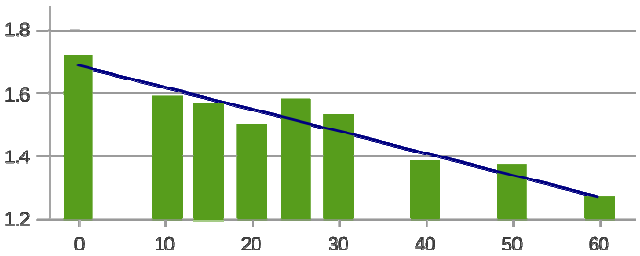


Figure 12. Coefficient of Building Compactness. X-axis – Resolution of the Generalization (0 - Original Buildings).

In Fig. 12 we can see that the coefficients of the buildings' compactness decreases significantly from 1.71 to 1.27, which demonstrates the efficiency of the approach.

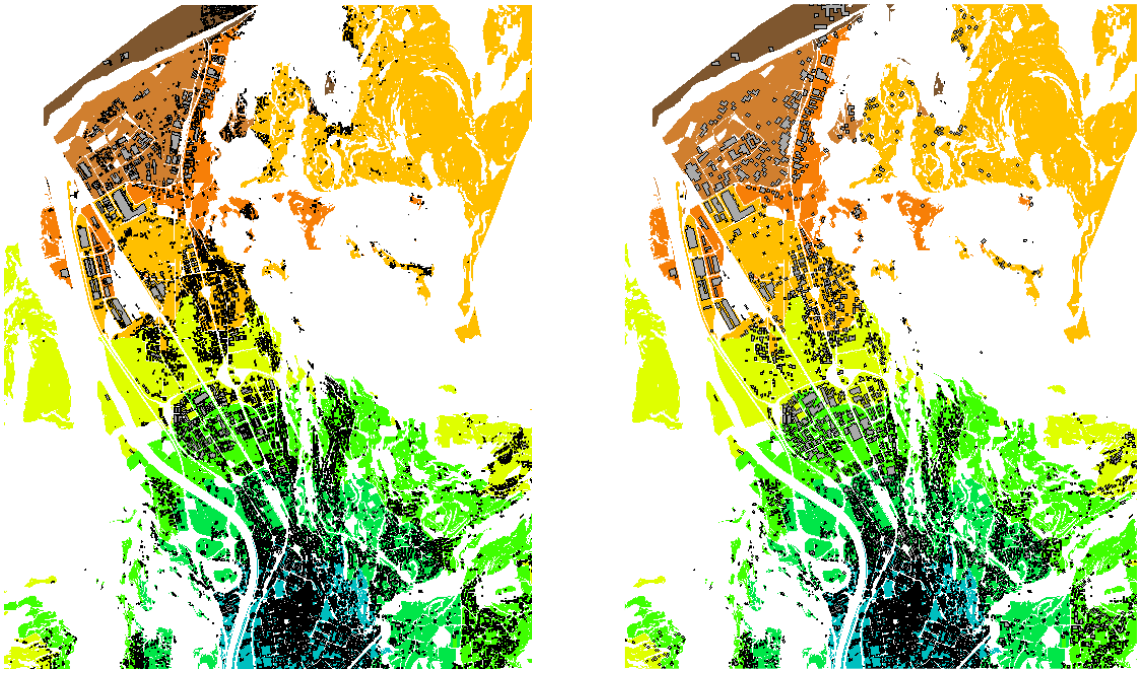


Figure 13. The Northern half of Trento with the Original Buildings (left) and with the Generalized Buildings (right): Different Levels of Generalization and Background Colors are according to Table I.

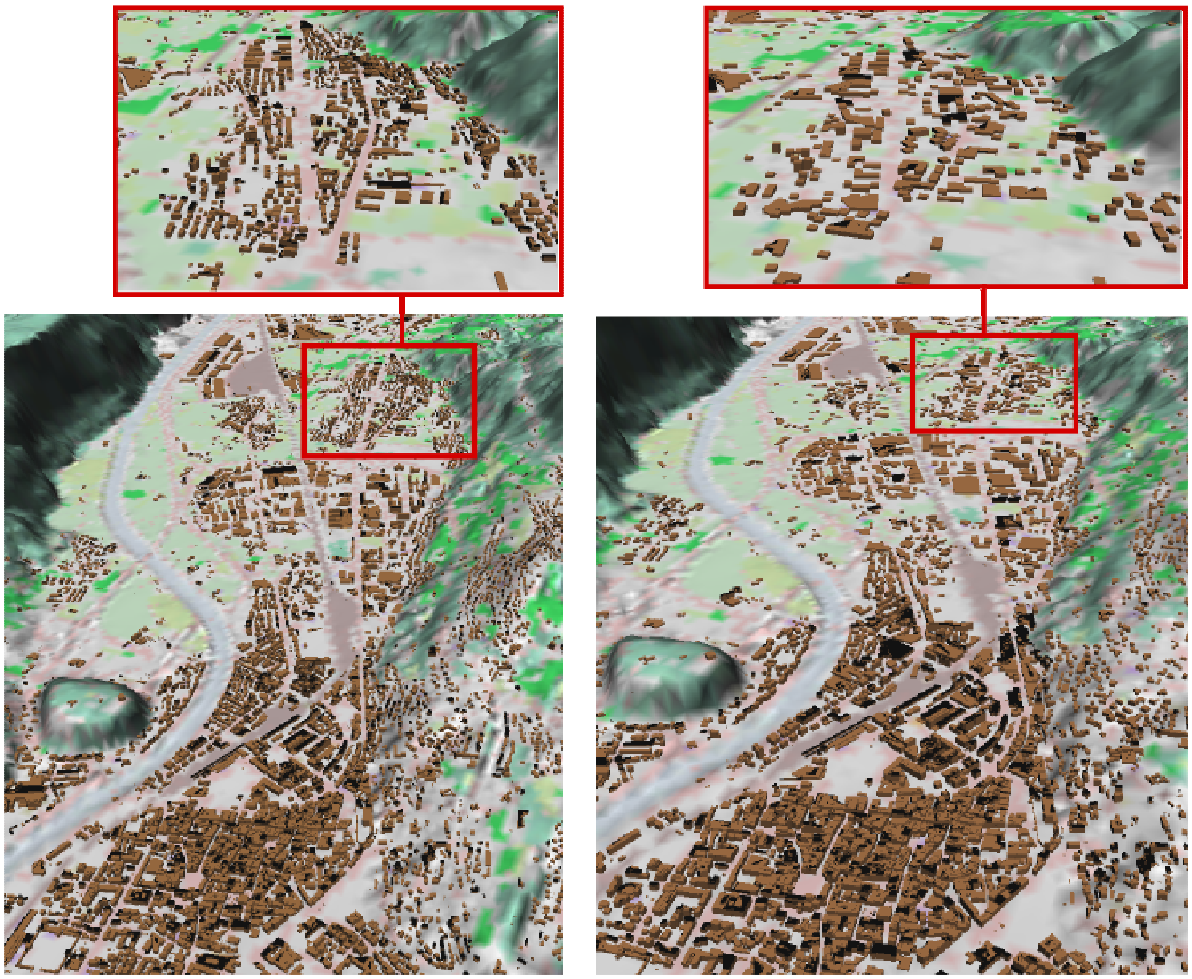


Figure 14. 3D Perspective with the Original Buildings (left) and with the Generalized Buildings (right). Zoomed Areas are Marked in Red.

The method and the process were developed by using a standard PC (DELL Vostro 3550), 4 processors: Intel® Core™ i3-2310M CPU @ 2.10GHz, with 1.8 GB Memory. In addition, Debian GNU/Linux 7 operating system, GRASS GIS, Bash and R programming languages were used.

VIII. CONCLUSION AND OUTLOOK

A new method for the 3D generalization of groups of buildings has been presented. To implement a multi-scale buildings generalization, a new hierarchical approach to the generalization of quarters and their contained buildings was developed. The approach is based on classification of quarters according to multiple attributes and on buffering operations. The raster-based approach of the method for buildings generalization is based on standard tools of rasterization, vectorization, region growing, and overlaying. The main advantage of the developed method is the ability to simplistically and efficiently generalize buildings at different levels, achieving variable, but continuous, level-of-detail of the buildings as a function of the depth of the plotted perspectives. The continuity of the generalized product is achieved by subdividing the area of the city into quarters, which take into account the significant objects affecting the process. As a result, the generalized 3D model does not contain unreadable and overly detailed separate buildings on the one hand, and is able to merge further groups of buildings on the other. At the same time, even though the buildings are simplified, the model maintains the geographical correctness and specifications of the urban area.

The developed method helps reduce the time, and the computer resources required, for drawing 3D models or perspectives of a city or urban areas.

REFERENCES

[1] Noskov A., Doytsher Y., “Urban Perspectives: A Raster-Based Approach to 3D Generalization of Groups of Buildings,” GEOProcessing 2013, pp. 67-72, France, 2013.

[2] Ankerst M., Breunig M., Kriegel H. P., Sander J, “OPTICS: ordering points to identify the clustering structure,” ACM SIGMOD Record, vol. 28, pp. 49-60, 1999.

[3] Breunig M., Zlatanova S., "3D geo-database research: Retrospective and future directions," Computers & Geosciences, vol. 37, 2011.

[4] Döllner J., Buchholz H., "Continuous Level-of-detail Modeling of Buildings in 3D City Models", in GIS'05 Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems, pp. 173-181, ISBN:1-59593-146-5, 2005.

[5] Ester M., Kriegel H. P., Sander J., Xu X, "A density-based algorithm for discovering clusters in large spatial databases with noise," KDD, vol. 96, 1996.

[6] Estivill-Castro V., "Why so many clustering algorithms: a position paper," ACM SIGKDD Explorations Newsletter, vol. 4, pp. 65-75, 2002.

[7] Forberg A., "Generalization of 3D Building Data Based on a Scale-Space Approach", ISPRS Journal of Photogrammetry & Remote Sensing, vol. 62: pp. 104-111, 2007.

[8] Glander T., Döllner J. “Abstract representations for interactive visualization of virtual 3D city models”. Computers, Environment and Urban Systems, vol. 33, 2009.

[9] Gröger G., Kolbe T.H., Plümer L., "City Geographic Markup Language", Approved Discussion Paper of the Open Geospatial Consortium, 2006.

[10] Guercke R. , Götzelmann T., Brenner C., Sester M. “Aggregation of LoD 1 building models as an optimization problem”. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 66, 2011.

[11] He S., Moreau G., Martin J. “Footprint-Based 3D Generalization of Building Groups for Virtual City”. GEOProcessing 2012 : The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services.

[12] Hildebrandt D., Döllner J, “Service-oriented, standards-based 3D geovisualization: Potential and challenges,” Computers, Environment and Urban Systems, vol. 34, pp. 484-495, 2010.

[13] Isikdag U., Zlatanova S., "Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models", in 3D Geo-Information Sciences Lecture Notes in Geoinformation and Cartography, Part II, pp. 79-96, DOI: 10.1007/978-3-540-87395-2_6, 2009.

[14] Joshi M., "Classification, Clustering and Intrusion Detection System," International Journal of Engineering Research and Applications, vol. II, pp. 961-964, 2012.

[15] Joubran J., Doytsher Y., "An Automated Cartographic Generalization Process: A Pseudo-Physical Model", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII, part B2, 2008.

[16] Kada M., "3D Building Generalization Based on Half-Space Modeling", Joint ISPRS Workshop on Multiple Representation, 2006.

[17] Kada, M., "Automatic Generalisation of 3D Building Models", in Proceedings of the Joint International Symposium on Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002.

[18] Kohonen T., “Self-organizing maps,” 3rd Edition, Springer, ISBN 3-540-67921-9, 2001.

[19] Kolbe T., "Representing and Exchanging 3D City Models with CityGML", in Proceedings of the 3rd International Workshop of 3D Geo-information, Seoul, Korea, 2009.

[20] Kolbe T.H., "CityGML – OGC Standard for Photogrammetry?", Photogrammetric Week, Stuttgart, Germany, 2009.

[21] Kolbe T.H., Gröger G., "Towards Unified 3D City Models", in Schiewe, J., Hahn, M, Madden, M, Sester, M. (Eds.): Challenges in Geospatial Analysis, Integration and Visualization II. Proceedings of Joint ISPRS Workshop, Stuttgart, Germany, 2003.

[22] Kolbe T.H., Gröger G., "Unified Representation of 3D City Models", Geoinformation Science Journal, vol. 4, 2004.

[23] Kolbe T.H., Gröger G., Plümer K., "CityGML – Interoperable Access to 3D City Models", in Proceedings of the First International Symposium on Fachbeiträge Geo-information for Disaster Management, Delft, The Netherlands, 2005.

[24] Kriegel, H., "Density-based clustering," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 1, pp. 231-240, 2011.

[25] Li Z., Yan H., Ai T. and Chen J. “Automated building generalization based on urban morphology and Gestalt theory”. Int. J. Geographical information science, vol. 18, 2004.

[26] Lloyd S.. "Least squares quantization in PCM," Information Theory, IEEE Transactions, vol. 28, pp. 129-137, 1982.

[27] Luebke D., Reddy M., Cohen J.D., Varshney A., Watson B., Huebner R., "Level of Detail for 3D Graphics (The Morgan Kaufmann Series in Computer Graphics)" ISBN: 978-1558608382, Edition 1, 2002.

[28] Mao B., Ban Y., “Online Visualization of 3D City Model Using CityGML and X3DOM,” Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 46, pp. 109-114, 2011.

[29] Over M., Schilling A., Neubauer S., Zipf A., “Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany,” Computers, Environment and Urban Systems, vol. 34, pp. 496-507, 2010.

[30] Prieto I., Izkara, J., “Visualization of 3D city models on mobile devices,” In Proceedings of the 17th International Conference on 3D Web Technology, pp. 101-104, 2012.

[31] Schilling A., Hagedorn B., Coors V., “OGC 3D Portrayal Interoperability Experiment Final Report”, Open Geospatial Consortium. OGC 12-075 (opengis.net/doc/ie/3dpie), 2012.

[32] Sharma N., Jain R., Yadav M., "Efficient and fast clustering algorithm for real time data," International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, vol. 2, 2012.

[33] Shiode N., "3D Urban Models: Recent Developments in the Digital Modelling of Urban Environments in Three-Dimensions", GeoJournal vol. 52, pp. 263-269, 2000.

[34] Shojaei D., Kalantari M., Bishop I. D., Rajabifard A., Aien A.. “Visualization requirements for 3D cadastral systems”, Computers, Environment and Urban Systems, vol. 41, pp. 39-54, 2013.

[35] Stadler A., Nagel C., König G., Kolbe T.H., "Making Interoperability Persistent: A 3D Geo Database Based on CityGML", 3D Geo-Information Sciences. Lecture Notes in Geoinformation and Cartography, Part II, 2009.

[36] Thiemann F., "Generalization of 3D Building Data", in Proceedings of Symposium of Geospatial Theory, Processing and Applications, Ottawa, Canada, 2002.

[37] Trapp M., Glander T., Buchholz H., "3D Generalization Lenses for Interactive Focus + Context Visualization of Virtual City Models", in Proceedings of the 12th International Conference Information Visualization, 2008.

[38] Uden M., Zipf, A., “Open Building Models: Towards a Platform for Crowdsourcing Virtual 3D Cities,” In Progress and New Trends in 3D Geoinformation Sciences, pp. 299-314, Springer, Berlin Heidelberg, , Germany, 2013.

[39] Wu H., He Z., Gong J. , “A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes”, Computers, Environment and Urban Systems, vol. 34, pp. 291-298, 2010.

[40] Xie J., Zhang L., Li J., "Automatic Simplification and Visualization of 3D Urban Building Models", International Journal of Applied Earth Observation and Geionformation, vol. 18, pp. 222–231, 2012.

[41] http://webapps.comune.trento.it/ambiente/ [accessed: 2013-02-12].

www.iariajournals.org

International Journal On Advances in Intelligent Systems

ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, ENERGY, COLLA, IMMM, INTELLI, SMART, DATA ANALYTICS

issn: 1942-2679

International Journal On Advances in Internet Technology

ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING, MOBILITY, WEB

issn: 1942-2652

International Journal On Advances in Life Sciences

eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO, SOTICS, GLOBAL HEALTH

issn: 1942-2660

International Journal On Advances in Networks and Services

ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION, VEHICULAR, INNOV

issn: 1942-2644

International Journal On Advances in Security

ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS

issn: 1942-2636

International Journal On Advances in Software

ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS, IMMM, MOBILITY, VEHICULAR, DATA ANALYTICS

issn: 1942-2628

International Journal On Advances in Systems and Measurements

ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL, INFOCOMP

issn: 1942-261x

International Journal On Advances in Telecommunications

AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA, COCORA, PESARO, INNOV

issn: 1942-2601