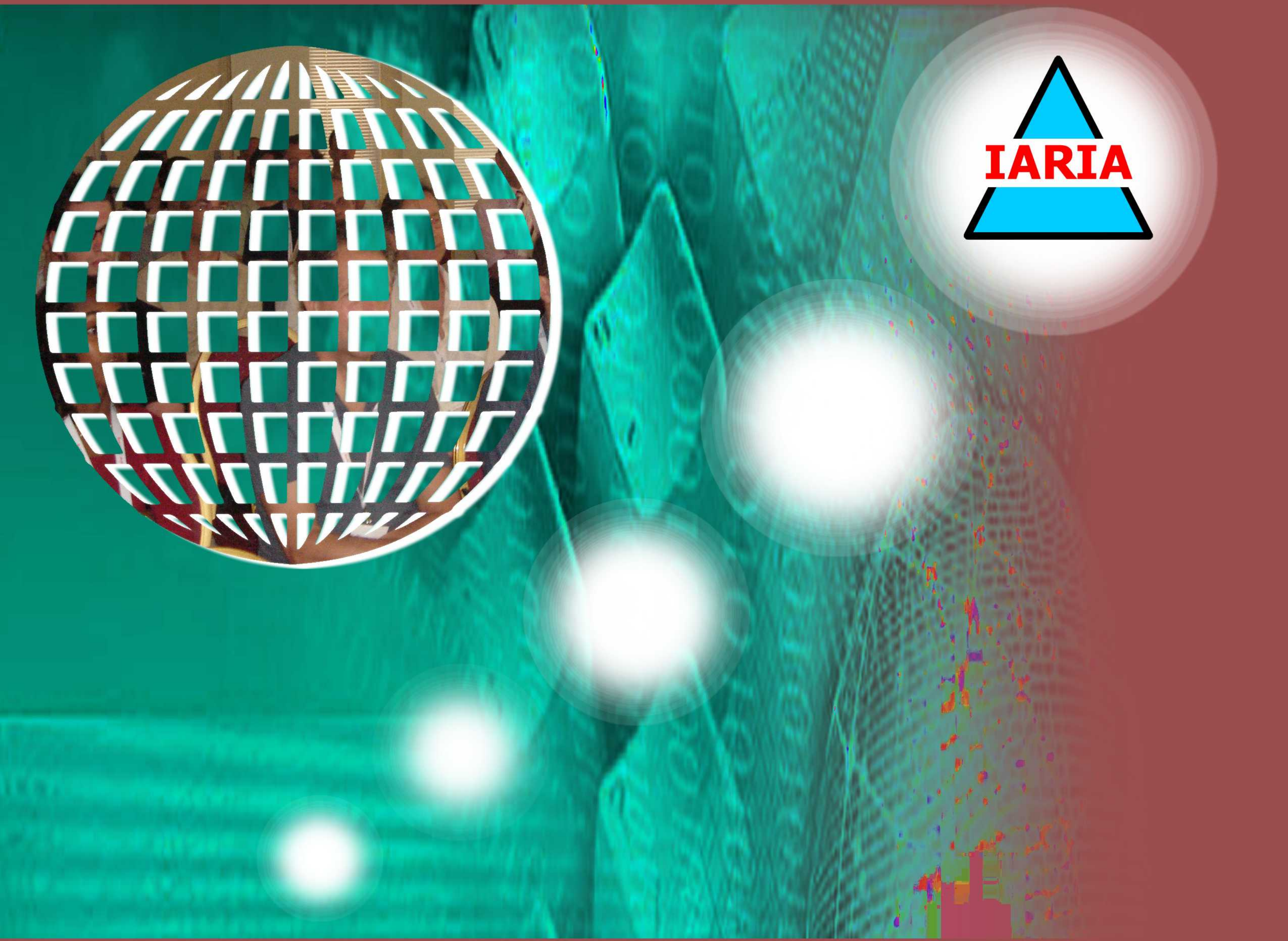
International Journal on Advances in Software



2009 vol. 2 nr. 2&3

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

International Journal on Advances in Software
Volume 1, Numbers 2&3, 2009

- Olga Ormandjieva, Concordia University-Montreal, Canada
- Constantin Paleologu, University ‘Politehnica’ of Bucharest, Romania
- Liviu Panait, Google Inc., USA
- Kenji Saito, Keio University, Japan
- Ashok Sharma, Satyam Computer Services Ltd – Hyderabad, India
- Marcin Solarski, IBM-Software Labs, Germany

Advanced Computing

- Matthieu Geist, Supelec / ArcelorMittal, France
- Jameleddine Hassine, Cisco Systems, Inc., Canada
- Sascha Opletal, Universitat Stuttgart, Germany
- Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
- Meikel Poess, Oracle, USA
- Kurt Rohloff, BBN Technologies, USA
- Said Tazi, LAAS-CNRS, Universite de Toulouse / Universite Toulouse1, France
- Simon Tsang, Telcordia Technologies, Inc. - Piscataway, USA

Geographic Information Systems

- Christophe Claramunt, Naval Academy Research Institute, France
- Dumitru Roman, Semantic Technology Institute Innsbruck, Austria
- Emmanuel Stefanakis, Harokopio University, Greece

Databases and Data

- Peter Baumann, Jacobs University Bremen / Rasdaman GmbH Bremen, Germany
- Qiming Chen, HP Labs – Palo Alto, USA
- Ela Hunt, University of Strathclyde - Glasgow, UK
- Claudia Roncancio INPG / ENSIMAG - Grenoble, France

Intensive Applications

- Fernando Boronat, Integrated Management Coastal Research Institute, Spain
- Chih-Cheng Hung, Southern Polytechnic State University, USA
- Jianhua Ma, Hosei University, Japan
- Milena Radenkovic, University of Nottingham, UK
- DJamel H. Sadok, Universidade Federal de Pernambuco, Brazil
- Marius Slavescu, IBM Toronto Lab, Canada
- Cristian Ungureanu, NEC Labs America - Princeton, USA

Testing and Validation

- Michael Browne, IBM, USA
- Cecilia Metra, DEIS-ARCES-University of Bologna, Italy
- Krzysztof Rogoz, Motorola, USA
- Sergio Soares, Federal University of Pernambuco, Brazil

- Alin Stefanescu, SAP Research, Germany
- Massimo Tivoli, Universita degli Studi dell'Aquila, Italy

Simulations

- Robert de Souza, The Logistics Institute - Asia Pacific, Singapore
- Ann Dunkin, Hewlett-Packard, USA
- Tejas R. Gandhi, Virtua Health-Marlton, USA
- Lars Moench, University of Hagen, Germany
- Michael J. North, Argonne National Laboratory, USA
- Michal Pioro, Warsaw University of Technology, Poland and Lund University, Sweden
- Edward Williams, PMC-Dearborn, USA

International Journal on Advances in Software
Volume 2, Numbers 2&3, 2009

CONTENTS

Dynamical Parallel Applications on Distributed and High Performance Computing Systems

(Invited Paper)

Claus-Peter Rückemann
Leibniz Universität Hannover (LUH), Hannover, Germany
Westfälische Wilhelms-Universität Münster (WWU), Münster, Germany
North-German Supercomputing Alliance (HLRN), Germany
Email: rueckemann@rrzn.uni-hannover.de

Abstract

This paper provides an extended overview of recent developments for exploiting distributed, Grid and High Performance Computing (HPC) resources with applications like Geoscientific Information Systems (GIS) based on the GISIG actmap-project. Focus is on frameworks for optimising the dynamical parallel use of computing resources for future cooperation and development concepts, integrating software and hardware architecture aspects. Using parallel processing and the method of event triggering from within Active Source can be used to exploit the vast computing power of distributed MultiCore parallel systems for a multitude of purposes like geoinformation processing, geophysical data analysis, information systems, and e-Science. An extended case study for an application InfoPoint demonstrates the algorithm here. As various obstacles showed up with implementing and minimising the complexity of the application-resource workflow, the creation of future Web and HPC services on top of HPC and Distributed Systems will be a solution for dedicated issues. The new extended framework of the Grid-GIS house is presented here, showing the case study for using these concepts for exploration purposes. For implementation testing distributed resources and the new multi-site supercomputer resources of HLRN-II (North-German Supercomputing Alliance) have been used.

Keywords

High Performance Computing; Distributed Systems; Grid-Computing; e-Science; Geoscientific Information Systems.

1. Introduction

The future of creating effective and efficient applications for dynamical visualisation and information systems is tightly linked with taking the advantage of parallel processing on MultiCore systems. Dynamical visualisation and advanced geoscientific information systems are prominent examples [1] at state of the art of development. Using Distributed Systems and High Performance Computing (HPC) resources therefore requires new concepts as integrating these resources, that in nearly all case do have an unique architecture and basic system configuration is a challenge for development and portability.

Extending the application spectrum, a new success story using InfoPoints (groups of active information objects) is presented implementing concepts of the Active Source framework for using distributed components and resources, suitable for Grid, Cloud, and HPC.

An extended implementation of the “Grid-GIS house” framework for building services on top of Distributed and HPC Systems for this purpose is presented here for the first time. Within the “Grid-GIS house” the state of the art in accounting and billing for has been considered for creating an integrated solution embracing all High End Computing (HEC) namely HPC and Cluster Computing as well as distributed and service oriented architectures with Grid Computing and Cloud Computing [2], [3]. At the state of the art of computing, hardware development today is getting near the physical limits and software development faces new challenges. This extended implementation is currently used for building interdisciplinary cooperations for the purpose of implementing geo-exploration systems based on parallel computing components. As the next generation of dynamical applications in the disciplines involved is as well strongly depending on backend software as on hardware components and high end networks this integrated modular framework has proven suitable.

In the last years strong interests emerged, regarding High End Computing like HPC and Distributed Computing, spanning industry as well as natural sciences [4], [5], [6]. HPC resources available with the North-German Supercomputing Alliance (HLRN) have been used for testing these developments. Software and hardware architecture are discussed as resources used in the future will have to be efficiently configured for the purpose of dynamic and interactive use.

Dynamical applications are characterised by the ability to present various information and context based on interaction in very flexible ways. The concept of Active Source and the

Object Graphics data type [7] based on source code, can be used to create and integrate such applications. An example is dynamical visualisation, being able to create complex dynamical charts and diagrams enriched with accompanying visual multimedia information, where the context can change the state of the view.

Dynamical applications are in many cases limited by the local resources as there can be large computational requirements for generating new dynamical views in a short time as for example with vertices calculation in Geoscientific Information Systems (GIS) or multimedia production on demand for Points of Interest (POI) data. Parallel programming can make use of distributed resources enabling thousands of parallel processes.

The combination of dynamical applications and parallel programming components lead to "Dynamical Parallel Applications" being able to use loosely coupled as well as embarrassingly parallel methods depending on the tasks.

Numerous applications and algorithms for handling dynamical visualisation and processing of scientific information could evolve even more flexibility and facilities if they could use existing computing power more directly, namely MPP (Massively Parallel Processing) and SMP (Symmetric Multi-Processing) resources. Large benefits can result from using many cores of large computing resources in parallel, within a shorter time interval, for quasi interactive use.

This paper presents the origins, problems and challenges (Section 2-4) as well as the status of the implementation (Section 5). A case study and a detailed InfoPoint example will illustrate this (Section 6-7). Issues on software and hardware resources used will be discussed, being essential for effective distributed applications in the future (Section 8). This leads to an evaluation and future work already begun with the extended service-oriented framework for Distributed and High Performance Computing (Section 9).

2. Origin and prior art

The idea of dynamical, distributed resource usage for geoscientific information was introduced with the concept of Active Source [7]. Over the years a Grid-GIS framework with many features had to be implemented within the GISIG actmap-project [8] including several programming libraries providing a suitable Application Programming Interface (API).

With computing resources evolving towards many cores [9], [10], [11], [12] the idea of using these systems more widely had been internationally presented and some major obstacles have been identified [13], [14]. For integration of HPC, Grid, and cluster resources these are:

- framework for the use of high end computing resources for dynamical visualisation and information systems,
- integrability of concepts (e.g. batch and scheduling),
- frameworks for the application of algorithms needed,
- interfaces for flexible and secure data and application transfer, interchange, and distribution,
- portability of implementations, extendability of existing methods, reusability of existing solutions.

Due to the limitations of "delivering" computing power from High Performance Computing, Grid Computing, and cluster computing resources interactively to a local application on some workstation, a framework is needed to integrate these resources. In absence of support for coupling these resources, in the past some features had to be last on the list to be addressed.

3. Problems addressed

As described in previous publications [13] GIS, Grid, and HPC are working on the GISIG implementation in order to overcome current obstacles, developing frameworks for the use of HPC and MultiCore computing resources, interfaces for data and application interchange, integrability, and portability.

This paper does proceed to implement and disseminate the proposed frameworks and interfaces for the purpose of demonstrating implementing ways for opening powerful High Performance Computing resources to specialised scientific applications and e-Science. It shows the first implementation results of case studies on a new HPC resource, using Massively Parallel Processing and Symmetric Multi-Processing components of the HLRN architecture with distributed resource locations.

Primary target disciplines are geoinformation processing, seismic processing for oil and gas, geophysical data analysis, computing expensive natural resource information systems, computational geology, hurricane tracking, dynamical cartography, and geostatistics.

4. Challenges identified

The most important challenges identified with these implementations on HPC resources have been grouped within this context in order to be briefly discussed.

- HPC resources and configuration,
- batch system and scheduling,
- accessing computing resources / Actmap Computing Resources Interface / Message Passing,
- distributing data,
- authorisation and system security,
- accounting jobs and processes.

The following sections briefly describe the basic approaches for the implemented solution before showing an overall case study of an information system using distributed resources.

5. Mastering complexity

As it has been shown, the problems we encounter with exploiting top-level backend resources are manyfold. Currently the systems available only provide access to local resources or there is only a batch implementation. In order to solve the problems with implementing a complex system, affords the integration of different technologies and methods:

- availability of Geoscientific Information System components at source level (actmap),
- support for event-driven dynamical applications (Active Source),
- access to High End Computing resources (SMP, MPP, MultiCore, Grid) and configuration,
- creation of loosely parallel coupling interfaces for interactive batch jobs,
- parallelisation of functional components and algorithms on High End Computing resources.

Emphasis has been put onto the integration of these topics. In practice experts from different disciplines, information sciences, geosciences, computer science, engineering, are engaged. The integration has been done by defining an open framework for this purpose, based und the Grid-GIS house.

6. Status of the implementation

For the work described here, various distributed and HPC resources of HLRN have been used. This is the derivative based on new complementary methods to the predecessor work handling Grid and Cluster Computing resources [15].

6.1. HPC resources and configuration

HLRN is the North-German Supercomputing Alliance. HLRN provides high-end High Performance Computing (HPC) resources jointly used and co-funded by the northern German states of Niedersachsen, Berlin, Bremen, Hamburg, Mecklenburg-Vorpommern, Schleswig-Holstein, and the Federal Government of Germany / German Research Society (DFG).

Those resources include HLRN-II [9], a system comprised of two identical computing and storage complexes, one located at the Leibniz Universität Hannover, Regionales Rechenzentrum für Niedersachsen (RRZN) and the other at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB). By connecting the two systems via the HLRN-Link dedicated fibre optic network (Cisco Catalyst switches), HLRN can operate and administer them as one system.

Each complex consists of MPP and SMP cluster components (SGI Altix ICE and XE) [16] installed in two phases. The first phase has been installed by Silicon Graphics Inc. in the year 2008.

The HLRN-II system (at 312 TFlop/s peak) operated with SLES is used by scientists for HPC applications from a wide range of disciplines, including Geosciences, Environmental Sciences, Climatology, Physics, CFD, Modeling and Simulation, Chemistry, Biology, and Engineering. All projects are supported by the HLRN service and competence network.

So while still in an early phase this resource installation, incorporating different computing components, gave the suitable context for individually configuring an implementation as described in the following sections. With the available HPC resources a number of software, application, and network components have been configured (Table 1) for integrating the framework and preparing a suitable software and hardware environment for the case study scenarios.

Component	*Software / Configuration*
Frameworks	GISIG, Actmap CRI, Grid-GIS
Operating System	S.u.S.E Linux / SLES
Batch system	Moab, Torque
Networks	MPI (InfiniBand), I/O (InfiniBand), service and administrative networks
Parallelisation	MPI, OpenMP, MPT, MPICH
Transfer / interchange	Secure Shell / keys, `pdsh`
Security	Sandboxing, Tcl, Tcl Plugin
Policies	`home`, `javascript`, `trusted`
Compilers	Intel Fortran, C, C++ suite, PGI, GNU
Libraries & Appl.	BLAS, LAPACK, NAG, ATLAS, CPMD, MOLPRO, FEOM, NAMD, Gaussian, FFT, TAU, NWChem, VMD, EnSight, ABAQUS, ANSYS, FLUENT, STAR-CD . . .
Parallelisation	SGI MPI / MPT, Intel MPI, OpenMP, MPICH, MVAPICH, SHMEM . . .

Table 1. HPC software components configured.

This is an excerpt of basic software components like applications, libraries and compilers, available for applications discussed in the context of this paper.

For security reasons a trusted computing interface using sandboxing has been configured as various security policies for integrating data and applications have been introduced and successfully tested.

This configuration allows very flexible transfer of data, secured execution of foreign Active Sources on demand, accounting as well as batch and interactive use of resources.

The basic trusted environment is independent from the computing architecture and can be used out of the box. The specific architecture dependent configuration part must be done accordingly to the purposes where it is neccessary for the service.

Primary targets might be key management, LDAP or firewall configuration. The components used for management of the components are shown in Table 2. Information on the current state of these resources can be found online [9].

Component	*Management / Configuration*
Additional Server	HOME, Data, Login, Batch, LDAP OSS, MDS, QFS, Repository
Mgmt. Altix ICE / XE	SGI Tempo / Scali Manage
Storage / Global FS	RAID 6, Storage Manager / Lustre
File Replication	NetVault Replicator (HLRN-Link)
Software Access	"Modules" (Compilers, Libs, Apps)
Monitoring	Nagios, Ganglia
Grid, Access, HA	Grid tools, Middleware, . . .
Profiling / Debugging	Intel Threading & Tracing Tools, PCP, PerfSuite, TotalView, ddt, gdb
Configuration Mgmt.	Cfengine, CVS

Table 2. HPC management components configured.

6.2. Batch system and scheduling

The batch system, scheduling and resource management implemented on HLRN-II is based on Moab and Torque. With this system the PBS (Portable Batch System) resource specification language [17] [18] is used. Interactive use and calculation is highly dependent on features of the batch system used. Currently the end user application will have to do the job synchronisation. With a conventional system configuration the management of multi user operation is difficult. Both synchronising and multi user operation tend to work against interactive use.

6.3. Accessing computing resources

The Actmap Computing Resources Interface (CRI) is an actmap library containing procedures for handling computing resources. Examples for using High Performance Computing and Grid Computing resources include batch system interfaces and job handling.

Listing 1 shows a simplified source code part of the actmap call for loading the Actmap Computing Resources Interface (Tcl or TBC) into the application stack. This library can be extended and modified interactively on the fly or via scripting [19].

```
#BCMT--------------------------------------------------
###EN \gisigsnip{Load actlcri}
#ECMT--------------------------------------------------
if {"$behaviour_loadlib_actlcri" == "yes"} {
  catch {
    if {[info exists tcl_platform(isWrapped)]} {
      puts "actlcri.tbc␣library␣initialized␣..."
      source actlcri.tbc
      set status_in_actmap yes
    } else {
      puts "actlcri.tcl␣library␣initialized␣..."
      source [file join $ACTMAPHOME "actlcri.tcl"]
      set status_in_actmap yes
    } } }
```

Listing 1. Calling CRI.

This library (`actlcri`) can hold functions and procedures and even platform specific parts in a portable way. It can be used by calling the source code library as well as the byte code library generated with a compiler like TclPro. From an application, calling Actmap CRI can be done as follows. For various applications, byte code (TBC) [7] has been considered for any part of applications and data.

With CRI being part of Active Source, parallel processing interfaces for Message Passing e.g. using InfiniBand, can be used, for example MPI (Message Passing Interface) and OpenMP. Listing 2 and Listing 3 show an MPI and an OpenMP job script used with Actmap CRI.

```
#!/bin/bash
#PBS -N myjob
#PBS -j oe
#PBS -l walltime=00:10:00
#PBS -l nodes=8:ppn=4
#PBS -l feature=ice
#PBS -l partition=hannover
#PBS -l naccesspolicy=singlejob
module load mpt
cd $PBS_O_WORKDIR
np=$(cat $PBS_NODEFILE | wc -l)

mpiexec_mpt -np $np ./dyna.out 2>&1
```

Listing 2. Active Source MPI (SGI MPT) script.

```
#!/bin/bash
#PBS -N myjob
#PBS -j oe
#PBS -A myproject
#PBS -l walltime=00:10:00
#PBS -l nodes=1:ppn=4
#PBS -l feature=xe
#PBS -l naccesspolicy=singlejob
cd $PBS_O_WORKDIR
export OMP_NUM_THREADS=4

./dyna.out 2>&1
```

Listing 3. Active Source OpenMP script.

Scripts of this type will on demand — this means using event binding — be sent to the batch system for processing. The sources can be semi-automatically generated, can be called from a set of files or can be embedded into an actmap component, depending on the field of application.

6.4. Distributing data

Within event triggered jobs, MPI and batch means can be used for distributing and collecting data and job output. For distributing files automatically within the system e.g. `dsh`, `pdsh`, C3 tools, Secure Shell (`ssh` and `scp`) are used. Interactive communication is supported by the appropriate Secure Shell key configuration. It must be part of the system configuration to correctly employ authorisation keys and `crontab` or `at` features.

6.5. Authorisation and system security

Authorisation for accessing data and information associated with the calculation currently affords to have one instance of the application present on one of the servers of the HPC resource, e.g. login or batch. A dedicated network using secure keys can be configured for the purpose of interactive application access in order to simplify communication and data transfer between the nodes. As for system security reasons large installations will tend to be restricted to dedicated users with this scenario. For execution of dynamic sources the trusted computing interface has been configured as `policy trusted`.

6.6. Accounting jobs and processes

The implemented framework is incorporated in an integrated solution for monitoring, accounting, billing supporting the geoinformation market. An outlook has been given for Geographic Grid Computing at the International Conference on Grid Services Engineering and Management (GSEM). Especially for the extended use of GIS and computing resources, the Grid-GIS framework, the “Grid-GIS house” has been created [13] and is used within the D-Grid [20], [21] and with Condor. The Active Source components used here, are part of this framework, on top of the Grid services, Grid middleware, and the HPC and Grid resources.

7. Case study

The selected case study overview shows different high level GIS views implemented with dynamical cartography (Active Map) in order to enable geocognostic insights. With this solution, processing, data storage, and information retrieval is done by using distributed resources. Handling is triggered from within the application by events via the Active Source framework. In oder to concentrate on the views we omit features previously demonstrated, such as active elements handling and visualisation, multimedia objects and raster and vector layering.

With a suitable interface, distributed computing resources can be used for creating any part of the application or data. So data collection and automation, data processing, and data transfer can be handled via existing means.

For example parallel processing of satellite data or satellite photos can be triggered from within the Active Map. The precalculation of views (Listing 4) can be automated from the application, processing several hundred views at a time using dedicated compute nodes for each calculation.

```
convert -scale 2400x1200 inview01.jpg outview01.jpg
convert -scale 2400x1200 inview02.jpg outview02.jpg
convert -scale 2400x1200 inview03.jpg outview03.jpg
...
```

Listing 4. Precalculation of satellite data.

An event binding command is shown in Listing 5. These bindings can bind events to selective objects of a category. The number of objects handled in object source is only limited by the system and hardware used. This way it is possible to provide any part of the application with support of distributed computing and storage resources, e.g. for simple cases via HTTP or HTTPS. The functional part can be a procedure, another component or an executable.

```
$w bind precalc_bio <Button-1> {exec precalc_bio.sh}
```

Listing 5. Binding of precalculation script.

For the following examples all the components are linked by the GISIG Active Source framework using event programming and the most computing intensive operations are done in the background on HPC compute resources.

Figure 1 shows part of an active satellite worldmap calculated on a HPC compute node as described. The respective action for calculating the view is linked into the Active Source data via an event bind call (Tcl) to the batch script. The batch script using scripting and MPI is executed by the batch system (Moab / Torque) to run on the compute nodes of the specified MPP component.

The result is transferred back to an application working directory from where the results calculated on the compute nodes are loaded into the active map (Tcl canvas) in order to build the desired view. Any objects of these views do get unique identification keys and may be automatically equipped with logical identification strings.

From within this interactive view one might want to switch to an active ocean / depth or plate tectonics view in a next step as in Figure 2 and end up in showing a vegetation / biology view as in Figure 3.

Once calculated all the maps exist at the same time, they can be regarded “precalculated”. Active Source uses a layer concept meaning any number of objects can be grouped in separate layers with all layers representing a stack of layers. It can be defined for the specific Active Source application if all of the calculated views do reside in memory, stacked in layers inside of the application as described or if they shall be removed in favour of releasing memory.

In the first case no data has to be recalculated, any views precalculated this way can be accessed interactively. It can be easily switched between the views by predefined events. The standard ways for doing so are key bindings to rotate views and mouse events to bring the next or a defined view to the front.

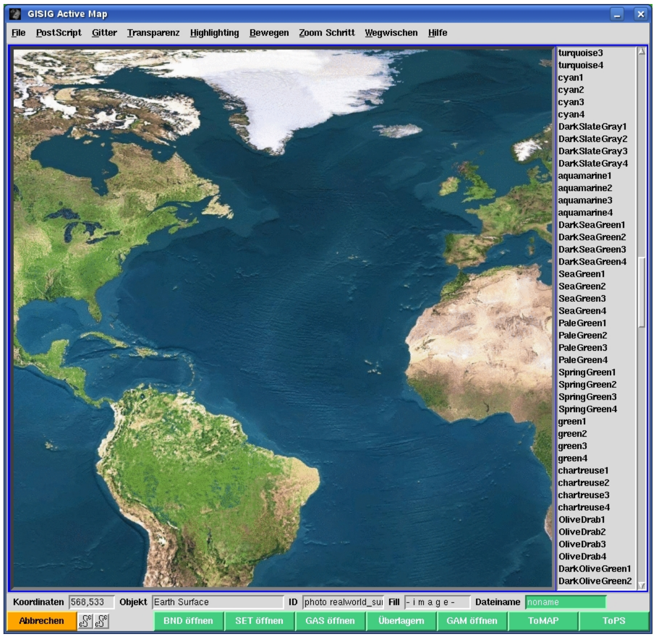


Figure 1. Precalculated topography view.

Any operation on the data suitable for interactive and batch mode can be done from within the Active Source framework. With this capability batch jobs can be created, e.g. for dynamically adding synthetic data and raytraced elements in interactive mode. Using parallelised applications like parallel POV-Ray from within these jobs, distributed computing resources can be used most effectively.

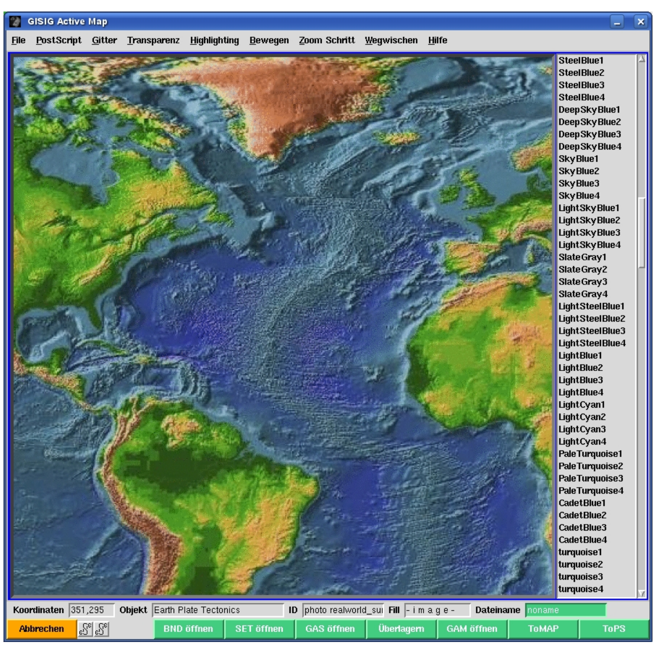


Figure 2. Precalculated plate tectonics view.

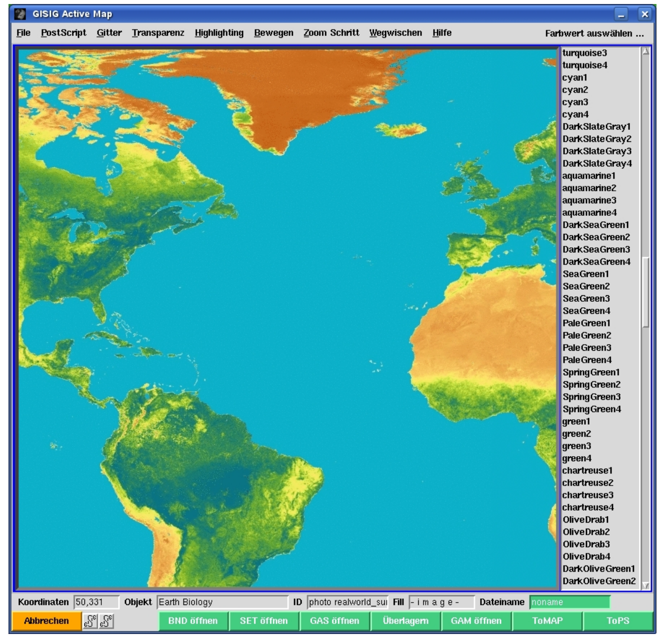


Figure 3. Precalculated biology view.

This can e.g. be encouraged in order to enhance geocognostic views by generating hundreds of data sets for points of interest. Ongoing from Figure 3, decision may fall to viewing the pollution distribution within a city in a distinct area as in Figure 4. One will select an “active spot” on the map that is linked with an appropriate detailed active city map. Most flexible geocognostic views can be created this way using the local and background computing resources at any time in the process of user interaction. GISIG Active Maps can consist of vector and raster layers as well as of multimedia components and events. Problems of dynamical cartography and geocognostic views with millions of data points having to be connected with live, interactive data being very computing intensive can be solved.

The example (Figure 4) shows a dynamical event-driven city map containing environmental and infrastructure data that is delivered from distributed sources. Now if one wants to take a look at pollution values of the largest lake within this city, as in Figure 5, a right click onto that object will display the results. Any interactive and batch events may be defined. A defined key bound will toggle a legend. Further zooming can be done to any extent, e.g. to resolve elementary objects within views. This demonstrates cartography combined with aerial data (vegetation and topography), and vector data (infrastructure and surfaces of water) all linked by events, and extensible by event triggered computing.

The selected part shown, is a highly zoomed area of the previously presented map, here in different thematical geocognostic context. Arbitrary detailed satellite maps and supporting data may be calculated on the HPC resources using the described algorithm.

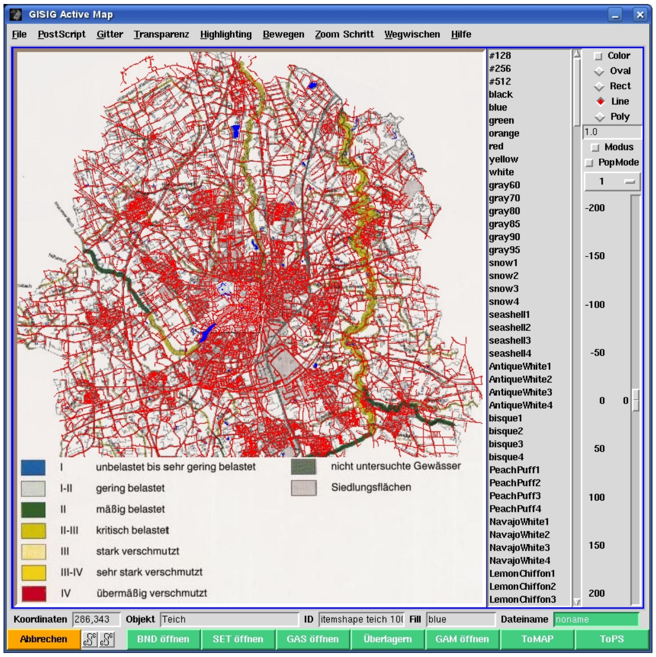


Figure 4. City, vector / raster layers, events.

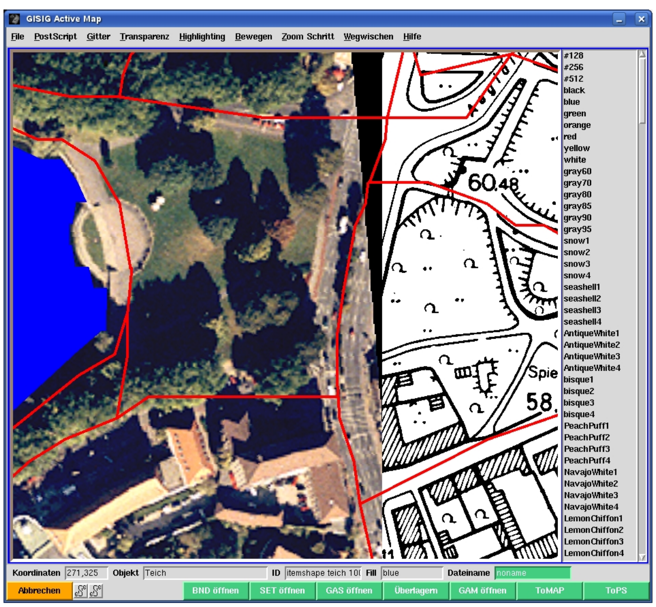


Figure 5. Detail, combined geocognostic view: map data, aerial data, and vector data.

Used from via a login node the solution with HPC compute nodes does show less latency than for previous solutions with distributed Grid resources. The login nodes used, are configured for interactive use of the batch system so there is no queue wait-time and much less time necessary for scheduling and re-scheduling. That way, avoiding a standard batch system configuration and a high job load, interactive applications are possible, reducing the wait times from about 12 up to 24 hours down to 1 to 10 minutes for medium expensive computation events. For less consumptive computation events the overall wait times are in the range of seconds.

e-Science applications like dynamical cartography and visualisation can use distributed resources in combination with Inter-Process Communication (IPC) and remote control within the application in a standard way as the parts calculated externally have been delivered back and loaded into the application. Any part can be reloaded or removed from memory separately so that memory usage is minimised.

8. InfoPoints using distributed resources

Using auto-events, dynamical cartography, and geocognostic aspects, views and applications using distributed compute and storage resources can be created very flexibly.

As with the concept presented resources available from Distributed Systems, High Performance Computing, Grid and Cloud services, and available networks can be used. The main components are:

- interactive dynamical applications (frontend),
- distributed resources, compute and storage, configured for interactive and batch use,
- parallel applications and components (backend), as available on the resources,
- a framework with interfaces for using parallel applications interactively.

Besides the traditional visualisation a lot of disciplines like exploration, archaeology, medicine, epidemology and for example various applications within the tourism industry can profit from the e-Science components. These e-Science components can be used for Geoscientific Information Systems for dynamical InfoPoints and multimedia, Points of Interest based on Active Source (Active POI), dynamical mapping, and dynamical applications.

8.1. InfoPoints and dynamical cartography

Figure 6 shows an interactive Map of México. The yellow circle is an event sensitive Active Source object containing a collection of references for particular objects in the application. This type of object has been named InfoPoint. InfoPoints can use any type of start and stop routines triggered by events. Figure 7 shows a defined assortment of information, a view set, fetched and presented by triggering an event on the InfoPoint.

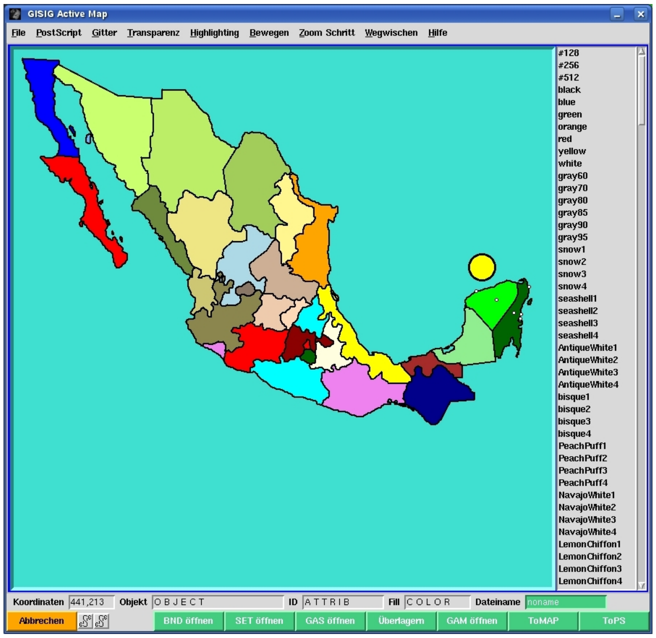


Figure 6. Interactive México with InfoPoint Yucatán.

Figure 7. Sample view set of InfoPoint Yucatán.

The information has been referenced from within the World Wide Web in this case. InfoPoints can depend on the cognitive context within the application as this is a basic feature of Active Source: Creating an application data set it is for example possible to define the Level of Detail (LoD) for zoom levels and how the application handles different kinds of objects like Points of Interest (PoI) or resolution of photos in the focus area of the pointing device.

8.2. Inside InfoPoints

The following passages show all the minimal components necessary for a fully functional InfoPoint. The example for this case study is mainly based on the Active Source framework. Triggered program execution ("Geoevents") of applications is shown with event bindings, start and stop routines for the data.

8.3. InfoPoints bindings and creation

Listing 6 shows the creation of the canvas for the InfoPoint and loading of the Active Source via bindings.

```
#
# actmap example -- (c) Claus-Peter R\"uckemann,␣2008,␣2009
#

#
# Active map of Mexico
#

erasePict
$w configure -background turquoise

pack forget .scale .drawmode .tagborderwidth \
  .poly .line .rect .oval .setcolor
pack forget .popupmode .optmen_zoom

openSource    mexico.gas
removeGrid

##EOF:
```

Listing 6. Example InfoPoint Binding Data.

This dynamical application can be created by loading the Active Source data with the `actmap` framework (Listing 7).

```
/home/cpr/gisig/actmap_sb.sfc mexico.bnd
```

Listing 7. Example creating the dynamical application.

8.4. InfoPoints Active Source

The following Active Source code (Listing 8) shows a tiny excerpt of the Active Source for the interactive Map of México containing some main functional parts for the InfoPoint Yucatán (as shown in Figure 6).

```
#BCMT-----------------------------------------------
###EN \gisigsnip{Object Data: Country Mexico}
###EN Minimal Active Source example with InfoPoint:
###EN Yucatan (Cancun, Chichen Itza, Tulum).
#ECMT-----------------------------------------------
proc create_country_mexico {} {
global w
#  Yucatan
$w create polygon 9.691339i 4.547244i 9.667717i \
  4.541732i 9.644094i 4.535433i 9.620472i 4.523622i \
  9.596850i 4.511811i 9.573228i 4.506299i 9.531496i \
  4.500000i 9.507874i 4.518110i 9.484252i 4.529921i \
  9.460630i 4.541732i 9.437008i 4.547244i 9.413386i \
  4.553543i 9.384252i 4.559055i 9.354331i 4.565354i \
  9.330709i 4.588976i 9.307087i 4.612598i 9.283465i \
  4.624409i 9.259843i 4.636220i 9.236220i 4.641732i \
  9.212598i 4.641732i 9.188976i 4.648031i 9.165354i \
  4.653543i 9.141732i 4.659843i 9.118110i 4.665354i \
```

```
  9.094488i 4.671654i 9.070866i 4.677165i 9.047244i \
  4.688976i 9.023622i 4.695276i 9.000000i 4.707087i \
  8.976378i 4.712598i 8.952756i 4.724409i 8.929134i \
  4.730709i 8.905512i 4.736220i 8.881890i 4.748031i \
  8.858268i 4.766142i 8.834646i 4.783465i 8.811024i \
  4.801575i 8.787402i 4.813386i 8.763780i 4.830709i \
  8.751969i 4.854331i 8.740157i 4.877953i 8.734646i \
  4.901575i 8.728346i 4.925197i 8.746457i 4.937008i \
  8.751969i 4.966929i 8.751969i 4.978740i 8.763780i \
  5.007874i 8.763780i 5.019685i 8.787402i 5.025984i \
  8.805512i 5.031496i 8.817323i 5.049606i 8.846457i \
  5.055118i 8.876378i 5.055118i 9.248031i 5.468504i \
  9.673228i 4.896063i 9.744094i 4.748031i 9.720472i \
  4.553543i \
 -outline #000000 -width 2 -fill green -tags {itemshape province_yucatan}
}

proc create_country_mexico_bind {} {
global w
$w bind province_yucatan <Button-1> {showName "Province␣Yucatan"}
$w bind province_quintana_roo <Button-1> {showName "Province␣Quintana␣Roo"}
}

proc create_country_mexico_sites {} {
global w
global text_site_name_cancun
global text_site_name_chichen_itza
global text_site_name_tulum
set text_site_name_cancun         "Cancún"
set text_site_name_chichen_itza   "Chichén␣Itzá"
set text_site_name_tulum          "Tulum"

$w create oval 8.80i 4.00i 9.30i 4.50i \
  -fill yellow -width 3 \
  -tags {itemshape site legend_infopoint}
$w bind legend_infopoint <Button-1> \
  {showName "Legend␣InfoPoint"}
$w bind legend_infopoint <Shift-Button-3> \
  {exec browedit$t_suff}

$w create oval 9.93i 4.60i 9.98i 4.65i \
  -fill white -width 1 \
  -tags {itemshape site cancun}
$w bind cancun <Button-1> \
  {showName "$text_site_name_cancun"}
$w bind cancun <Shift-Button-3> \
  {exec browedit$t_suff}

$w create oval 9.30i 4.85i 9.36i 4.90i \
  -fill white -width 1 \
  -tags {itemshape site chichen_itza}
$w bind chichen_itza <Button-1> \
  {showName "$text_site_name_chichen_itza"}
$w bind chichen_itza <Shift-Button-3> \
  {exec browedit$t_suff}

$w create oval 9.76i 5.20i 9.82i 5.26i \
  -fill white -width 1 \
  -tags {itemshape site tulum}
$w bind tulum <Button-1> \
  {showName "$text_site_name_tulum"}
$w bind tulum <Shift-Button-3> \
  {exec browedit$t_suff}
}

proc create_country_mexico_autoevents {} {
global w
$w bind legend_infopoint <Any-Enter> {set killatleave [exec ./mexico_legend_infopoint_viewall.sh $op_parallel ] }
$w bind legend_infopoint <Any-Leave> {exec ./mexico_legend_infopoint_kaxv.sh }

$w bind cancun <Any-Enter> {set killatleave [exec $appl_image_viewer -geometry +800+400 ./mexico_site_name_cancun.jpg $op_parallel ] }
$w bind cancun <Any-Leave> {exec kill -9 $killatleave }

$w bind chichen_itza <Any-Enter> {set killatleave [exec $appl_image_viewer -geometry +800+100 ./mexico_site_name_chichen_itza.jpg $op_parallel ] }
$w bind chichen_itza <Any-Leave> {exec kill -9 $killatleave }

$w bind tulum <Any-Enter> {set killatleave [exec $appl_image_viewer -geometry +800+400 ./mexico_site_name_tulum.jpg $op_parallel ] }
$w bind tulum <Any-Leave> {exec kill -9 $killatleave }
}

proc create_country_mexico_application_ballons {} {
global w
global is1
gisig:set_balloon $is1.country "Notation␣of␣State␣and␣Site"
gisig:set_balloon $is1.color "Symbolic␣Color␣od␣State␣and␣Site"
}

create_country_mexico
create_country_mexico_bind
create_country_mexico_sites
create_country_mexico_autoevents
create_country_mexico_application_ballons

scaleAllCanvas 0.8
##EOF
```

Listing 8. Example InfoPoint Active Source data.

The source contains a minimal example with the active objects for the province Yucatán in México. The full data set contains all provinces as shown in Figure 6. The functional parts depicted in the source are the procedures for:

- `create_country_mexico`:
  The cartographic mapping data (polygon data in this example only) including attribute and tag data.
- `create_country_mexico_bind`:
  The event bindings for the provinces. Active Source functions are called, displaying province names.
- `create_country_mexico_sites`:
  Selected site names on the map and the active objects for site objects including the InfoPoint object. The classification of the InfoPoint is done using the tag `legend_infopoint`. Any internal or external actions like context dependent scripting can be triggered by single objects or groups of objects.
- `create_country_mexico_autoevents`:
  Some autoevents with the event definitions for the objects (Enter and Leave events in this example).
- `create_country_mexico_application_ballons`:
  Information for this data used within the Active Source application.
- Call section: The call section contains function calls for creating the components for the Active Source application at the start of the application, in this case the above procedures and scaling at startup.

Any number of groups of objects can be build. This excerpt only contains Cancun, Chichen Itza and Tulum. A more complex for this example data set will group data within topics, any category can be distinguished into subcategories in order to calculate specific views and multimedia information, for example for the category `site` used here:

- `city` (México City, Valladolid, Mérida, Playa del Carmen),
- `island` (Isla Mujeres, Isla Cozumel),
- `archaeological` (Cobá, Mayapan, Ek Balam, Aktumal, Templo Maya de Ixchel, Tumba de Caracol),
- `geological` (Chicxulub, Actun Chen, Sac Actun, Ik Kil),
- `marine` (Xel Há, Holbox, Palancar).

Objects can belong to more than one category or subcategory as for example some categories or all of these as well as single objects can be classified `touristic`.

The data, as contained in the procedures here (mapping data, events, autoevents, objects, bindings and so on) can be put into a database for handling huge data collections.

8.5. Start an InfoPoint

Listing 9 shows the start routine data (as shown in Figure 7). For simplicity various images are loaded in several application instances (`xv`) on the X Window System. Various other API calls like Web-Get `fetchWget` for fetching distributed objects via HTTP requests can be used and defined.

```
xv -geometry +1280+0   -expand 0.8 mexico_site_name_cancun_map.jpg &
xv -geometry +1280+263 -expand 0.97 mexico_site_name_cancun_map_hotels.jpg &
xv -geometry +980+0    -expand 0.5 mexico_site_name_cancun.jpg &
xv -geometry +980+228  -expand 0.61 mexico_site_name_cancun_hotel.jpg &
xv -geometry +980+450  -expand 0.60 mexico_site_name_cancun_mall.jpg &
xv -geometry +980+620  -expand 0.55 mexico_site_name_cancun_night.jpg &
xv -geometry +740+0    -expand 0.4 mexico_site_name_chichen_itza.jpg &
xv -geometry +740+220  -expand 0.8 mexico_site_name_cenote.jpg &
xv -geometry +740+420  -expand 0.6 mexico_site_name_tulum_temple.jpg &
#xv -geometry +740+500  -expand 0.3 mexico_site_name_tulum.jpg &
xv -geometry +740+629  -expand 0.6 mexico_site_name_palm.jpg &
```

Listing 9. Example InfoPoint event start routine data.

8.6. Stop an InfoPoint

Listing 10 shows the stop routine data. For simplicity all instances of the applications started with the start routine are removed via system calls.

```
killall -9 --user cpr --exact xv
```

Listing 10. Example InfoPoint event stop routine data.

Using Active Source applications any forget or delete modes as well as using Inter Process Communication (IPC) are possible.

9. Software and hardware resources used

For using High Performance Computing (HPC) and Grid Computing resources (ZIVGrid, ZIVcluster, ZIVsmp, HLRN) for Distributed Computing with Geoscientific Information Systems (GIS) it is has been shown [13], [1], [2] to be necessary carrying out an integration and configuration regarding software and hardware components.

For the HPC resources it is an ongoing research and development goal to optimise the single-system-properties with the software and hardware installation used for the case studies discussed within this paper. Several software / hardware configurations have been tested with the complex multi-cluster-multi-site installation of HLRN-II in order to ensure that the resulting system will be seen as one single system for system administration and various user applications.

9.1. Integrating SW and HW resource components

As the HLRN consists of two complexes located at two sites one goal is, to enable operation and use of all resources as one single system. The integration of the different SMP and MPP systems into this concept is an essential part, so accessing these resources via applications will be managed with an uniform interface. On the other hand it shall be possible to use the redundancies of the complexes to increase availability and minimise overall maintenance downtimes as with the system architecture it has been taken care that each complex can be down for full maintenance separately. The most important aspects of the single-system-properties in this context regard:

- Joint user and job management for one uniform user space, regarding an uniform addressing, use, and administration.
- System-spanning home directories including mirroring and replication, reducing the need for explicit data transfer and data synchronisation.
- Joint job and data scheduling with automated data transfer (data staging).
- Storage integration, integration of SAN capabilities, Data-Grid.
- MPI communication for very large applications using MPI-2 in user space in order to use resources of the spatially distributed complexes.

For the complexity involved with this, the following sections focus on the architecture and the hardware and software components and applications that had to be configured with the installation.

Currently application use cases have been internationally presented for this installation from application view only. The example use cases and most important results on hardware and software configuration are referenced in a separate section. This paper concentrates on the hardware and software components that had to be configured for creating a suitable integrated HPC resource base.

9.2. Architecture and phases

Table 3 gives a compact overview of the most important hardware resources of the HLRN-II system available for applications within the main installation phases. For the specific areas of application from the usage spectrum, the complexes consist of suitable SMP and MPP components. Separate InfiniBand networks for fast MPI and IO are available. The system complexes at the sites Hannover and Berlin are extended in two phases, with identical components and configuration. The hardware configuration details left out here due to legal issues will be provided in phase 2 of the installation process.

As with computing at the top edge of maximum performance and minimum of obstacles the story is not all about software only. There is a number of limits reducing efficiency that are immanently appearent with the architecture, in theses cases ordered by priority for use with the examples presented here:

- latencies of network and batch system limiting the response times for interactive use,
- throughput / IO limiting the streaming facilities with model calculation on the compute nodes and data servers,
- scalability of existing algorithms for computational problem solving,
- memory limiting efficient high-resolution simulation,
- storage capability limiting chain job restart and check-pointing,
- number of CPUs (cores) limiting the number of loosely coupled highly parallel compute events,
- availability of resources due to competitive jobs for different user applications,
- non-certified components limiting flexibility with application porting and configuration.

As a detailed description for a hardware solution is out of scope of this paper, the most important aspects for the applications handled are latencies and throughput. Fast dedicated networks for example using InfiniBand fabrics can help to reduce the bottlenecks and latencies for highly parallel as well as for dynamical and interactive applications. For example with event triggered "dynamically" changing visualisation controlled from within an interactive information system, large computation tasks as well as large visualisation IO (several hundred megabytes per second per task) can result. This will even increase in the near future. As far as separate physical networks dedicated for MPI and IO are available, applications will profit.

10. Evaluation and lessons learned

The current work of implementing and configuring software components and the case study shows use of computing resources with the Active Source framework, spatial event handling, and cognitive dynamical application.

With this solution it is possible to build sets of interactive, extensible, portable, and reusable applications with interdisciplinary background based on the computing power of MultiCore and HPC Systems.

In the last years many "flavours" of High End Computing have been evaluated. Summing up the experiences of the longterm project regarding this aspect, applications on the following architectures and paradigms have been successfully implemented and tested:

- Distributed and High Performance Computing (DHPC) on MPP, SMP, and vector computers,
- Grid Computing and Distributed Computing,
- Cluster Computing,
- Mobile, Utility, Tool, and Ubiquitous Computing.

With the current plans, the next topic on the agenda will be the Cloud Computing top service level – XaaS (Application as a Service, AaaS; Software as a Service / Security as a Service, SaaS) based on the base levels (Infrastructure as a Service, IaaS; Platform as a Service, PaaS; Desktop as a Service, DaaS).

The InfoPoint concept has been demonstrated, working for various disciplines, visualising and extending various features of cartography and e-Science under cognostic aspects. These applications may also use resources interactively but any short latencies are difficult to achieve with most current computing installations.

For optimising the use of resources the software configuration will have to be coordinated with the hardware configuration in order to build an efficient system architecture. Although the Active Source framework can integrate various concepts, it is highly dependent on the system configuration. The most obvious obstacles limiting efficiency and ease of use are the current state of HPC environments and the missing standardisation and modularisation of system components like for the batch system and scheduling. As in the HPC world every installation comes with an unique configuration, this is a crucial point. So always not only take a look on the software side but on the hardware, too.

11. Future work

The topics in focus for the next years can be grouped in three sections: technical aspect, collaboration work, and work within the participating disciplines.

11.1. Technical aspects

The basic algorithms have been implemented and tested for enabling distributed and HPC systems for dynamical use.

HLRN-II Overview	*Phase 1 (2nd Quarter 2008)*	*Phase 2 (from 2009 on)*	*Total*
Complex H/B each, MPP	MPP 1: SGI Altix ICE 8200EX (ICE+)	MPP 2: SGI Carlsbad 2	
Number of nodes (blades)	320 (Colfax-S w/ Seaburg)	960	1280
Number of sockets / cores	640 (Quad-Core) / 2560	[details provided in phase 2]	
Processor	Intel Xeon Harpertown, 3 GHz / 80 W	[Intel Next Generation Xeon]	
Memory & network	5.1 TByte (2 GByte/core, IB 4×DDR)	29.3 TByte (IB 2×Dual DDR)	34.4 TByte
System peak performance	30.7 TFlop/s	≈100 TFlop/s	≈130 TFlop/s
Complex H/B each, SMP	SMP 1: SGI Altix XE 1300	SMP 2: SGI UltraViolet	
Number of nodes	47 CN (+2 HN, XE250)	136	183
Number of sockets / cores	94 (Quad-Core) / 376	[details provided in phase 2]	
Processor	Intel Xeon Harpertown, 3 GHz / 80 W	[Intel Next Generation Xeon]	
Memory & network	2.8 TByte (8 GByte/core, IB 4×DDR)	8.7 TByte (NumaLink 5)	11.5 TByte
System peak performance	4.2 TFlop/s	≈22 TFlop/s	≈26 TFlop/s
Complex H+B overall	***Phase 1***	***Phase 2***	***Total***
Storage capacity (gross)	1.15 PByte (RAID-Array)	1.15 PByte (RAID-Array)	2.3 PByte
IO bandwidth	14 GByte/s	14 GByte/s	28 GByte/s
Number of cores (CN)	5824	19360	25184
Memory	16 TByte	76 TByte	92 TByte
System peak performance	70 TFlop/s	≈242 TFlop/s	≈312 TFlop/s

Table 3. HPC hardware resources in test situation, HLRN-II complexes Hannover (H) and Berlin (B).

The necessary configuration of systems and resources has to be standardised for practicing a uniform setup and in order to minimise invasive overhead. In the future it cannot be the user having the need to trigger most of the configuration of complex system components on every system an application should be run, there will have to be suitable interfaces.

There will have to be standard interfaces for parallelisation in the future. For both distributed and High Performance Computing, monitoring and accounting is necessary in order to handle interactive use.

The application of the frameworks presented for high level research and development consortium has already begun and will accelerate to develop standardised means of communication, like Web Services for HPC services for dedicated issues.

Currently the collaboration partners prepare to integrate the methods presented here for using distributed resources developed into components of open and commercial geoscientific information systems for productive use.

11.2. Collaboration work

Based the current organisational structure for combining work of the different interest groups, the block diagram in Figure 8 illustrates the future directions of integrating and co-developing large collaborative target frameworks and applications for service-oriented Distributed and High Performance Computing on management level. It shows the dependencies of

- market and services (green colour, shingle and cross pattern),
- computing services (red colour, brick pattern),
- HPC and distributed resources (blue colour, gravelly pattern),
- and resources to be provisioned or developed (yellow colour).

The collaboration partners in the fields of HPC, services, geosciences and exploration, do regard the modular three level framework structure essential for future development of an integrated solution.

As presented during the DigitalWorld conference 2009 in Cancún, México and with the Leadership in Research consortium, the proposed Computing Industry Alliance has been regarded to be a suitable umbrella organisation for Distributed and High Performance Computing and geoexploration sciences. The framework described is an example currently building the base for creating efficient interdisciplinary industry research cooperations for implementing the next generation of dynamical applications on Distributed and High Performance Computing resources based on the "Grid-GIS house" [13]. Interests to force this development exist not only in the Gulf of México region but as well in Russia and Saudi Arabia.

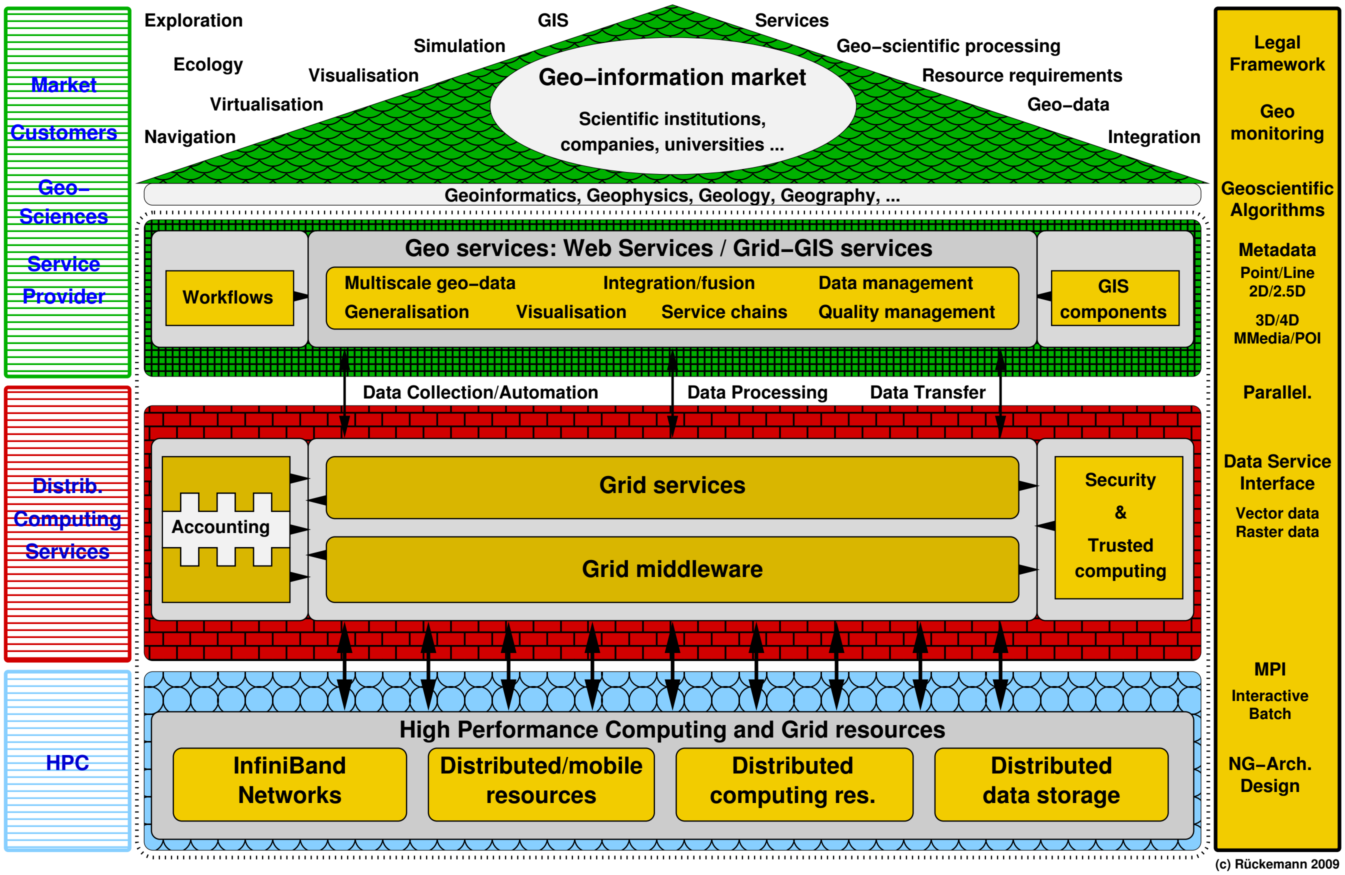


Figure 8. Future directions for service-oriented Distributed and High Performance Computing ("Grid-GIS house").

11.3. Disciplines

Three key player collaboration sections from High Performance Computing and Distributed and Grid Computing, from services and technical development, and from Geosciences are currently building the next generation of information and computation system as shown in Figure 8.

- For the HPC and distributed resources section top level (blue) HPC computing companies are engaged. Next generation architectures and standards, for example hardware and network configuration, batch, and MPI, for using, accessing, and managing backend resources are the most prominent goals. Cooperations like DEISA [22] and PRACE [23] expedite the evolution and visibility of the core factors for the overall European resources.
- For Distributed Computing services, Grid and Cloud (red) various organisations and activities regarding services and technology will be important [24], [25], [26], [27], [28], [29], [30], [31]. A number of requirements regarding Security are exposed to be handled in interdisciplinary context [32], [33], [34], [35], [36]. For building a market ready network of partners a flexible accounting is most important. Regarding accounting, an integrated solution with complex accounting units suitable for this scenario has been proposed [2] considering suitable components [37], [38], [39], [40].
- On the level of market and services (green) various key players cover science and research, as for geosciences and exploration. A lot of work has been done in the previous years in the disciplines of geophysics, seismics, seismology als well as regarding oil and gas in order to exploit High End Computing resources [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. The work has already begun on parallelisation of geoscientific algorithms for parallel processing. The future work will bring the essentials of these disciplines together in order build an information and computing system for the exploration sciences.

Currently there are no comprehensive frameworks available, directly comparable to the Grid-GIS house. On this top level for the next years, legal as well as technical aspects are most important for integration of national an international geospatial data integration (GDI / SDI) frameworks like GSDI, INSPIRE, GDI-DE, GMES, GEOSS and Public Sector Information (PSI) into these concepts.

12. Summary and concluding remarks

In this article the implementation and employment of dynamical applications for use with Distributed and High Performance Computing resources has been presented. The concept relies on source code based scripting applications for utilising computing resources for specialised information systems and e-Science. Event-driven object graphics are based on Active Source, which has been developed within the GISIG actmap-project.

Based on the current framework, efficient access to distributed computing resources from HPC to Grid Computing can be achieved. Design and configuration in most cases of HEC has to consider the hardware and network components, too. Standardising interfaces helps to simplify the problems of resource usage and encourage developers and users to build new parallel networking applications. Overcoming these obstacles using Distributed and HPC resources for dynamical application, the step currently done is to implement platforms with commercial support for integrating these features into future applications.

The higher-level result is, that it will only be possible to accomplish the goal of a flexible integrated information system for geosciences and exploration using distributed High End Computing resources if partners from computing, services, and various geoscience disciplines will collaborate. With this goal and based on the extended Grid-GIS house, building an high end international information computing system for the exploration sciences is currently under way.

Acknowledgements

We are grateful to all the colleagues and staff at the North-German Supercomputing Alliance (HLRN, Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen), at the Regionales Rechenzentrum für Niedersachsen (RRZN), Leibniz Universität Hannover (LUH), at the Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), and at Silicon Graphics Inc. (SGI) for their great work, support, and cooperation in the fields of operating, managing, and consulting of High Performance Computing resources within the North-German Supercomputing Alliance.

We like to thank all the colleagues at the Institut für Rechtsinformatik (IRI) and the European Legal Informatics Study Programme (EULISP) at the LUH, at the Westfälische Wilhelms-Universität Münster (WWU), the Leibniz-Rechenzentrum der Bayerischen Akademie der Wissenschaften, München (LRZ), at the Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH) of the Technische Universität Dresden (TUD), at the Forschungszentrum Jülich (FZJ), at the Forschungszentrum Karlsruhe (FZK), at the Hochleistungsrechenzentrum Stuttgart (HLRS), the Zentrum für Informationsverarbeitung Münster (ZIV), at the Research Center L3S, Hannover, the German Grid Initiative D-Grid, and at numerous other sites and HPC alliances, the Gauss Centre for Supercomputing (GCS), DEISA, PRACE, NorduGrid, and the AK Supercomputing / Zentren für Kommunikation und Informationsverarbeitung in Lehre und Forschung e.V. (ZKI) for the productive cooperation and information exchange over the last years.

Thanks to IBM, Sun, and SGI and all their staff members involved for supporting this work by managing and providing high end High Performance Computing, Cluster, Cloud, and Grid Computing resources and services for generations.

References

[1] C.-P. Rückemann, "Using Parallel MultiCore and HPC Systems for Dynamical Visualisation," in *Proceedings of the International Conference on Advanced Geographic Information Systems & Web Services (GEOWS 2009), February 1–7, 2009, Cancun, Mexico / ICDS 2009, ACHI 2009, ICQNM 2009, GEOWS 2009, eTELEMED 2009, eL&mL 2009, eKNOW 2009 / DigitalWorld 2009, International Academy, Research, and Industry Association (IARIA).* IEEE Computer Society Press, IEEE Xplore Digital Library, 2009, pp. 13–18, Dragicevic, S., Roman, D., Tanasescu, V. (eds.), 6 pages, BMS Part Number: CFP0973F-CDR, ISBN: 978-0-7695-3527-2, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4782685&isnumber=4782675 (PDF), URL: http://www.user.uni-hannover.de/cpr/x/bib/Rueckemann_2009_MultiCore_HPC.bib (BIBTEX entry), URL: http://www.iaria.org/conferences2009/AwardsGEOWS09.html (Best Paper Award) (HTML).

[2] C.-P. Rückemann, *Accounting and Billing in Computing Environments*, M. Pankowska, Ed. Business Science Reference, IGI Global, Hershey, Pennsylvania, USA, Oct. 2009, chapter X, in: Pankowska, M. (ed.), Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology, 421 pages, ISBN: 978-1-60566-890-1, URL: http://www.igi-global.com/reference/details.asp?ID=34799 (Information) (HTML), URL: http://www.igi-global.com/bsr/details.asp?ID=34799&v=preface (Preface) (HTML), URL: http://www.user.uni-hannover.de/cpr/x/bib/Rueckemann_2009_Computing_Environments.bib (BIBTEX entry).

[3] M. Pankowska, Ed., *Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology (in press).* IGI Global, Hershey, Pennsylvania, USA, Oct. 2009, ISBN: 978-1-60566-890-1, URL: http://www.igi-global.com/reference/details.asp?ID=34799 (Information) (HTML), URL: http://www.igi-global.com/bsr/details.asp?ID=34799&v=preface (Preface) (HTML), URL: http://www.user.uni-hannover.de/cpr/x/bib/Rueckemann_2009_Computing_Environments.bib (BIBTEX entry).

[4] "HPCwire," 2009, URL: http://www.hpcwire.com.

[5] "insideHPC," 2009, URL: http://www.insidehpc.com.

[6] "LX Project," 2009, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/de/index.html.

[7] C.-P. Rückemann, "Beitrag zur Realisierung portabler Komponenten für Geoinformationssysteme. Ein Konzept zur ereignisgesteuerten und dynamischen Visualisierung und Aufbereitung geowissenschaftlicher Daten," Dissertation, Westfälische Wilhelms-Universität, Münster, Deutschland, 2001, 161 (xxii + 139) Seiten, Ill., graph. Darst., Kt., URL: http://wwwmath.uni-muenster.de/cs/u/ruckema/x/dis/download/dis3acro.pdf.

[8] C.-P. Rückemann, "Active Map Software," 2001, 2005, URL: http://wwwmath.uni-muenster.de/cs/u/ruckema (information, data, abstract).

[9] "HLRN, North-German Supercomputing Alliance (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen)," 2009, URL: http://www.hlrn.de.

[10] "D-Grid, The German Grid Initiative," 2009, URL: http://www.d-grid.de.

[11] "D-Grid-Integrationsprojekt (DGI)," 2009, URL: http://dgi.d-grid.de.

[12] "NorduGrid," 2009, URL: http://www.nordugrid.org.

[13] C.-P. Rückemann, "Geographic Grid-Computing and HPC empowering Dynamical Visualisation for Geoscientific Information Systems," in *Proceedings of the 4th International Conference on Grid Service Engineering and Management (GSEM), September 25–26, 2007, Leipzig, Deutschland, co-located with Software, Agents and services for Business, Research, and E-sciences (SABRE2007)*, R. Kowalczyk, Ed., vol. 117. GI-Edition, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik e.V. (GI), 2007, 66–80 pages, ISBN: 78-3-8579-211-6, ISSN: 1617-5468.

[14] "Applications with Active Map Software, Screenshots," 2005, URL: http://wwwmath.uni-muenster.de/cs/u/ruckema/x/sciframe/en/screenshots.html.

[15] "GSEM, International Conference on Grid Services Engineering and Management," 2007, URL: http://www.gsem.de.

[16] "HLRN-II Photo Gallery," 2008, RRZN Top-News, May 9, 2008, URL: http://www.rrzn.uni-hannover.de/hlrn_galerie.html.

[17] "Torque Administrator Manual," 2009, URL: http://www.clusterresources.com/wiki/doku.php?id=torque:torque_wiki.

[18] "Moab Admin Manual, Moab Users Guide," 2009, URL: http://www.clusterresources.com/products/mwm/moabdocs/index.shtml, URL: http://www.clusterresources.com/products/mwm/docs/moabusers.shtml.

[19] "Tcl Developer Site," 2009, URL: http://dev.scriptics.com/.

[20] C.-P. Rückemann, Ed., *Ergebnisse der Studie und Anforderungsanalyse in den Fachgebieten Monitoring, Accounting, Billing bei den Communities und Ressourcenanbietern im D-Grid*. Koordination der Fachgebiete Monitoring, Accounting, Billing im D-Grid-Integrationsprojekt, 1. Juni 2006, D-Grid, Deutschland, 2006, 141 Pages, URL: http://www.d-grid.de/fileadmin/dgi_document/FG2/koordination_mab/mab_studie_ergebnisse.pdf.

[21] C.-P. Rückemann, W. Müller, and G. von Voigt, "Comparison of Grid Accounting Concepts for D-Grid," in *Proceedings of the Cracow Grid Workshop, CGW'06, Cracow, Poland, October 15–18, 2006*, M. Bubak, M. Turała, and K. Wiatr, Eds., Jul. 2007, pp. 459–466, ISBN: 83-915141-7-X.

[22] "Distributed European Infrastructure for Supercomputing Applications (DEISA)," 2009, URL: http://www.deisa.org.

[23] "Partnership for Advanced Computing in Europe (PRACE)," 2009, URL: http://www.prace-project.eu.

[24] "European Grid Initiative (EGI)," 2009, URL: http://www.egi.org.

[25] "Open Grid Forum (OGF)," 2009, URL: http://www.ofg.org.

[26] "Global Grid Forum (GGF)," 2009, URL: http://www.gridforum.org.

[27] "Globus Alliance," 2009, URL: http://www.globus.org.

[28] "UNICORE," 2009, URL: http://www.unicore.eu.

[29] "Tcl Developer Site," 2009, URL: http://dev.scriptics.com.

[30] "Building scalable, high performance cluster/grid networks: The role of ethernet," Force10 Networks: Cluster/Grid Computing, 2005, URL: http://www.force10networks.com/applications/roe.asp.

[31] "International Conference on Grid Services Engineering and Management (GSEM)," 2007, URL: http://www.gsem.de.

[32] A. Chakrabarti, *Grid Computing Security*, 1st ed. Springer Berlin Heidelberg New York, 2007.

[33] "Computer Emergency Response Team (CERT)," 2007, URL: http://www.cert.org.

[34] "KES – Die Zeitschrift für Informationssicherheit," 2007, URL: http://kes.info.

[35] "BSI-Forum," 2007, URL: http://www.bsi.bund.de/literat/forumkes.htm.

[36] C. Eckert, *IT-Sicherheit, Konzepte – Verfahren – Protokolle*, 4th ed. Oldenbourg Wissenschaftsverlag GmbH, München, Wien, 2006.

[37] E. Elmroth, P. Gardfjäll, O. Mulmo, Å. Sandgren, and T. Sandholm, *A Coordinated Accounting Solution for SweGrid*, Oct. 2003, URL: http://www.pdc.kth.se/grid/sgas/docs/SGAS-0.1.3.pdf.

[38] P. Gardfjäll, *Design Document: SweGrid Accounting System Bank*, Dec. 2003, URL: http://www.pdc.kth.se/grid/sgas/docs/SGAS-BANK-DD-0.1.pdf.

[39] P. Gardfjäll, E. Elmroth, L. Johnsson, O. Mulmo, and T. Sandholm, "Scalable Grid-wide capacity allocation with the SweGrid Accounting System (SGAS)," *Concurrency and Computation Practice and Experience*, 2006, (Submitted for Journal Publication, October 2006), URL: http://www.cs.umu.se/~elmroth/papers/sgas_submitted_oct_2006.pdf.

[40] J. Illik, *Electronic commerce: Grundlagen und Technik für die Erschließung elektronischer Märkte*, 2nd ed. Oldenbourg Wissenschaftsverlag GmbH, München, Wien, 2002.

[41] D. Bevc, O. Feodorov, and A. Popovici, "Internet-Based Processing: A Paradigm Shift for Exploration," in *Offshore Tech-*

nology Conference, Houston, TX, U. S. A., May 2001, URL: http://www.3dgeo.com/docs/OTC13276.pdf.

[42] D. Bevc, O. Feodorov, A. Popovici, and B. Biondi, "Internet-based seismic processing: The future of geophysical computing," in *70th Ann. Internat. Mtg., Soc., Expl. Geophys.*, Calgary, Canada, Aug. 2000, URL: http://www.3dgeo.com/docs/insp.pdf.

[43] "Internet enabling remote processing," The American Oil and Gas Reporter, 2001, URL: http://www.3dgeo.com/docs/0701_3dgeodevelopment_72dpi.pdf.

[44] F. Karbarz, "Grid computing for seismic processing," *The Leading Edge*, vol. 22, no. 1, pp. 58–60, 2003.

[45] DataSynapse, "Self-managed, guaranteed, distributed computing platform," *Industry White Paper: Enabling Government/Public Sector Applications for On-Demand Computing*, p. 28, 2003, URL: http://www.datasynapse.com/pdf/DataSynapse_Enabling_Government.pdf.

[46] D. Bevc, O. Feodorov, I. Musat, and S. Zarantonello, "Grid computing for energy exploration and development," in *Minisymposium on Grid Computing for the Oil and Gas Industry, SIAM Conference on Parallel Processing for Scientific Computing*, San Francisco, U. S. A., Feb. 2004, URL: http://www.3dgeo.com/docs/SIAM_3DGeo_abstract.pdf.

[47] D. Bevc, S. Zarantonello, N. Kaushik, and I. Musat, "Grid computing helps allocate scientific data," p. 14, URL: http://search390.techtarget.com/tip/1,289483,sid10_gci882336,00.html.

[48] "Sun Infrastructure Solution for Grid Computing: Oil and Gas," Sun, May 2004, URL: http://www.sun.com/solutions/documents/solution-sheets/EN_grid-oil+gas-ds_FF.xml.

[49] D. Thomas and M. Petitdidier, "EGEODE: a Grid Infrastructure for Research in Geosciences," *EAGE 67th Conference & Exhibition, Madrid, Spain, 13–16 June 2005*, 2005, URL: http://www.cgg.com/corporate/research/articles/eage05/Thomas.pdf.

[50] D. Bevc, S. Zarantonello, N. Kaushik, and I. Musat, "Grid computing for energy exploration," in *GGF14 - The Fourteenth Global Grid Forum*, Chicago, IL, U. S. A., 2005, URL: http://www.3dgeo.com/docs/GGF14_3DGeo.pdf.

[51] M. Käser, H. Igel, J. de la Puente, B. Schuberth, G. Jahnke, and P. Bunge, "Geowissenschaften: Erdbebenforschung durch Höchstleistungsrechnen, Moderne Supercomputer ermöglichen die Simulation realistischer Erdbeben-Szenarien unter Berücksichtigung komplizierter, geophysikalischer Erdmodelle," *Akademie Aktuell, Zeitschrift der Bayerischen Akademie der Wissenschaften*, vol. 02, pp. 47–50, 2006.

[52] A. Bachem, H.-G. Hegering, T. Lippert, and M. Resch, "The Gauss Centre for Supercomputing," *inSiDE, innovatives Supercomputing in Deutschland*, vol. 4, no. 2, pp. 4–5, Autumn 2006.

Educational Content Creation and Sharing in a Technology-rich Environment

Bernd J. Krämer and Peng Han
FernUniversiät in Hagen
Department of Mathematics and Computer Science
58084 Hagen, Germany
Email: {han.peng|bernd.kraemer}@fernuni-hagen.de

***Abstract*— The componentization and reuse of topical information and the organization of learning processes according to pedagogical theories have long been discussed separately in e-learning literature. The former was led under the buzzword "learning object", the latter focussed on learning activity sequencing and culminated in the Learning Design standard. This paper sketches a methodological framework and an e-learning portal that reconcile both strands of discussion in a comprehensive support in digital learning content production, adaptation and reuse. It presents an approach towards developing and reusing interactive learning objects relying on software design principles and adaptation mechanisms such as late composition and parameterization. Topical information like facts, concepts, procedures, processes or principles of a knowledge domain can be flexibly combined with learning objectives and activities supporting the learning process of an individual or group of learners. It suggests keeping information and educational context separate at design time and connecting both facets of learning objects only at reuse time. Parameterization is a software design principle used here to facilitate the adaptation of a learning object to different themes and didactic contexts. These design principles are illustrated for Java applets and for interactive Flash animations. This paper also illustrates facilities to adapt predefined didactic scenario templates, design new scenarios and update them with reusable learning objects from a repository or from the author's workspace.**

***Index Terms*— Learning object; configurable learning object; cognitive taxonomy; didactic scenario, didactic parameterization; content reuse; IPR; licensing.**

I. Introduction

The concept of learning objects arose in the early nineties driven by the motivation to reduce the development and maintenance cost of digital learning resources through modularization and reuse. Learning objects promised to offer a new way to create and mediate educational content in terms of smaller units of learning. These units are self-contained, can be reused in multiple contexts and different educational settings, and can be combined into coherent collections of learning materials. If didactically well designed, interactive learning objects can help students to understand comprehensive concepts and the inner working of complex processes better than from mere textual descriptions and static figures. This is particularly acute in self-paced learning situations, in which interactive multimedia learning objects can stimulate higher-level cognitive skills by allowing students to carry out procedures, to organize components of concepts or virtual materials, or even create new solutions.

The design and implementation of interactive learning objects is, however, time-consuming and requires special skills. Learning objects are also typically localized and tightly connected with particular didactic scenarios. However, this strongly limits their reuse in different contexts.

A. The CampusContent Project

In the main body of this paper, we present some findings made and results produced in the CampusContent project. CampusContent[1] is a competence center for e-learning that has been funded between March 2005 and July 2009 by the Deutsche Forschungsgemeinschaft[2]. The project was motivated by the observation that although a huge number of digital learning materials has been developed in the last decade, availability and access to these resources are limited and the degree of reuse is disappointingly low. The German Federal Minister of Education and Research, for example, invested millions of Euro at the beginning of this century in a four-year funding program called "New Media in Education", which aimed at the production of high quality digital learning content in and for German universities. However, the plethora of educational content resulting from such projects is difficult to find; it is not sustainably managed, and rarely has been designed for adaption and reuse in different learning arrangements. In contrast to books and scholarly journals that are systematically catalogued, managed and cross-referenced by libraries, no widely accepted archiving system and indexing standard exist that enable the systematic and effective storing, acquisition, distribution, and easy exchange of digital learning materials and representations of successful applications of didactic models.

CampusContent began its research at this point with the goals of:

- Reshaping the reuse and adaptability of digital learning objects to different application contexts,
- implementing and evaluating reference materials that illustrate design-for-reuse principles for learning objects,
- enabling experienced teachers and instructional design experts to represent best practices in teaching and learning and communicate these to practitioners, and

[1] `http://www.campuscontent.de/`

[2] DFG, the German Research Foundation, provided financial support under code number 44200719.

- supporting the work of course designers, teachers, and students through a coherent infrastructure that enables content sharing across heterogeneous learning management systems.

Later in project, we learned that the inclusion of social networking and collaboration functions could help users to organize communities of practice autonomously, furnish them with collective knowledge spaces and use functions for expressing recommendations, annotations and evaluations.

Typical use scenarios for the project's vision include:

- Author A uses resources from author B and author C, modifies them if licensing conditions permit, and adds her own content or didactic concepts to a seamless composition;
- author B and author C use the same material but for different instructional purposes or in different learning settings;
- a group of like-minded professors establishes a social network, e.g., on the topic "Service-Oriented Computing" and sets up a peer review system for learning materials on this topic;
- didactic experts represent online and blended learning models and didactic scenarios as learning paths or learning designs and publish them.

B. Resolving the ROI Paradox

Learning materials that can be used in different application contexts must be target-group and context neutral. However, good learning content should also be didactically tailored to the actual learning situation and learner group. Baumgartner named this conflict of goals the ROI (Reusability of Objects and Instruction) paradox [3].

We propose to mitigate the inherent contradiction between context-neutral content and the necessity of tailoring learning objects to the needs of the learner by a heuristic principle (see also [5]). This principle is known from software engineering as *late composition*. Adapted to e-learning, it suggests keeping information and didactic context separate at design time and connecting both facets of learning objects only at reuse time.

Fig. 1. Facets of a learning object

The project's model of a learning object was first published in [4]. A *learning object* combines an information object with a didactic scenario and a specific learning objective (see Fig. 1). An *information object* consists of illustrations, pieces of text, simulations, animations, video or audio clips, photos, maps, quizzes, reference works etc. that describe facts, concepts, procedures, processes or principles of a knowledge domain. A *didactic scenario* specifies roles and recommended learning or assessment activities, including learner-learner, learner-tutor, and learner-computer interactions. A *learning objective* specifies the skill development or knowledge acquisition anticipated as the result of a learning process. It connects the information object with the actual didactic scenario.

The components of a learning object are maintained sustainably as relational structures in the repository network the project has built. They can be retrieved and will inspire new combinations and adaptations in community processes, as we hope.

Parameterization is a mechanism also adapted from software engineering. We distinguish two forms: pedagogic and thematic parameterization. *Pedagogic parameterization* aims to equip information objects with parameters that allow its adaption to specific didactic needs. Besides other means, didactic parameterization can be used to realize late composition. We propose a scalar classification of learning objectives relying on Anderson and Krathwohl's taxonomy of cognitive processes [2] (see Section III). *Thematic parameterization* refers to the idea that certain interactive learning objects can be adapted to different topic areas by configuring a set of parameters.

In this article, we illustrate the implementation of these design principles and mechanisms for three types of learning objects that proved to be useful in higher education. We evolved these resources into generic objects from which custom-designed objects can be generated through combination, parameter configuration, and adaptation. Our first example, which is implemented in Java, serves to demonstrate the multitude of combinations we can achieve through late composition and didactic parameterization. Two further examples of generic objects are implemented in Adobe's Flash format. The first one, concept classification, serves to illustrate thematic parameterization, while the second Flash example illustrates the *separation-of-concerns* principle by which different features of an object like graphics design, interaction control, and functionality are treated separately.

C. Portal Edu-Sharing

Besides the conceptual and methodological results presented in this article, CampusContent developed a comprehensive portal, *Edu-Sharing*, that enables the sharing and reuse of digital learning content across heterogenous learning management systems. Versioning of content is supported.

Figure 2 depicts the core components and tools of Edu-Sharing. They can be grouped into authoring and learning support. The heart of the portal is a repository, in fact, a network of repositories because individual institutions may want to operate their own instance of an Edu-Sharing repository. Different instances of the portal repository can be connected through web services to form a distributed network providing a single system view from each participating site.

The distributed repository serves to organize and maintain personal workspaces of registered users and the outcome of authoring activities or content that is uploaded from the user's hard disk. Open interfaces allow different portal operators to connect their preferred authoring tools and learning management systems, while the repository component is standard to ensure interoperability in the network. A range of special editors and two open source learning management systems (LMSes), Moodle [27] and metacoon [24], are included in the standard distribution of Edu-Sharing. Plans and agreements with platform developers exist to interface further LMSes with Edu-Sharing.

The editors serve to produce or compose:

- Different types of data, such as text, video or graphic files representing basic building blocks of information and learning objects,
- assessment questions and tests conforming to the Question and Test Interoperability (QTI 2.0) standard [16],
- learning paths and didactic scenarios,
- learning objectives,
- learning objects, and
- course modules.

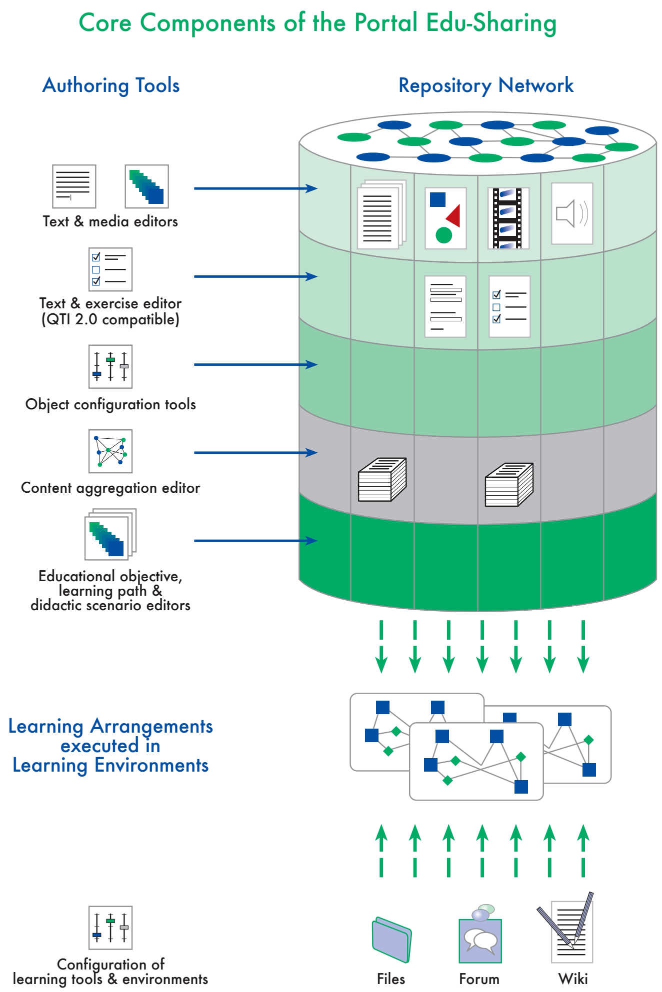


Fig. 2. Components of the portal Edu-Sharing

The repository network also supports learning processes performed in learning management systems or virtual learning environments directly because learning objects used in learning paths or pedagogical scenarios are referenced and executed from within the repository, as a rule. Learning objects can also be downloaded to execute a local copy. However, then the teacher loses the option to request usage data for his or her private instance from the portal's data analysis component (not shown in Fig. 2). Besides the possibility to reference learning objects from the repository network, the integration of repository and LMS offers further options on the LMS side: Search content in the repository; link or insert a learning path, didactic scenario, information or learning object found into the course under construction; store content built in the LMS sustainably in the repository. A license management component, which is also not shown, supports content owners upon upload to associate an appropriate use license with their works in the repository network. The personal workspace of each registered user serves to organize and connect clusters of documents and, more importantly, to share these with others, independent of usage rights and licenses. Thus, the workspaces provide a collaborative environment for communities of practice whose members have similar profiles or build on special trust relationships.

The portal was particularly designed to encourage the sharing and reuse of open educational resources [28]. It builds on the open source content management system Alfresco [1] and the open source portal software Liferay [23]. Currently Edu-Sharing undergoes a pilot use phase with different kinds of user groups at universities, schools and vocational training institutions [22]. By the end of 2009, the software packages developed in the project will be published as open source software to the public at large.

D. Structure of the Article

The article is an invited extension of a paper that was accepted for the International Conference on Mobile, Hybrid, and On-line Learning 2009 [13].

In the following section, we first report on related work. In Section III, we briefly review a well-known educational taxonomy dealing with cognitive aspects of learning. Then we show for a widely used class of models of computation, finite automata, how content can be flexibly combined with learning tasks addressing different levels of cognitive challenges. Section IV explains the components and architecture of the technology supporting our methodology for Flash-enabled objects. This section presents two examples of generic objects. In Section V we sketch an extension of our architecture that aims at raising the degree of adaptability of generic objects through a software component approach. Section VI presents some thoughts about design-for-reuse principles. Section VII explains how prerequisite requirements, learning and assessment activities, learning content, and completion requirements can be combined to learning paths and study courses. Section VIII finally touches upon IPR-related legal issues and discusses how they are addressed in the portal Edu-Sharing. We conclude with a brief summary and an outlook on future work.

II. Related Work

The reuse of digital learning material has been a continuing issue. First, there were a number of initiatives promoting the reuse of educational software. However, their success in practice was limited. The most substantial problems were incompatibilities in language, culture, curriculum, computer-use practices, and didactic approaches of the potential learners and their instructors [9].

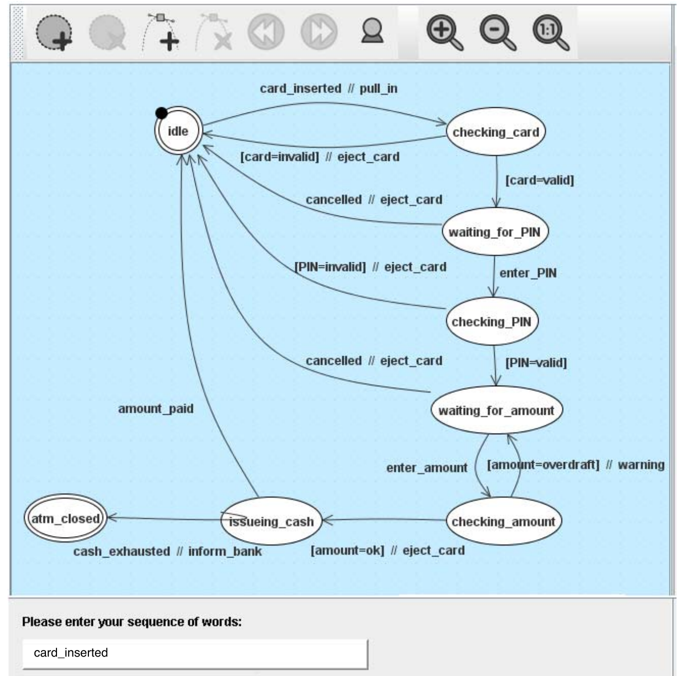
Although David Wiley compared the idea of building educational content from smaller building blocks with object-oriented programming [31], there is no generally agreed development and reuse concept as it exists, for instance, in software engineering. [21] argues that design principles such as encapsulation, cohesion, and decoupling, which allow software developers to develop and maintain objects independently of each other, should be carried over to learning objects to achieve similar benefits.

Boyle was the first who attempted to transfer certain software engineering principles like cohesion and decoupling to learning objects to encourage the production of reusable learning objects [7]. Cohesion among different components of a compound learning object in Boyle's approach is achieved by the fact that all components are focused on a single learning objective. IOs and learning activities with dynamic objects are combined to didactically purposeful learning objects. However, this technology only support white box reuse because a re-user who wants to change a compound object, has to manipulate it with a specific editing tool. More recently, in [18], Jones and Boyle adapted the design pattern approach [11] to learning objects. But this work is less concrete than what we propose in this article. A separation into content and didactic context in the sense of [14] to enhance a learning resource's reuse potential has not been practiced much.

III. Didactic Parameterization of Educational Resources

David Wiley seems to be the first who discussed the connection between learning objects and instructional design [31]. To achieve a practicable solution that seamlessly integrates the information and instructional facets of learning objects, we studied various educational taxonomies including Benjamin Bloom's well-known taxonomy of educational objectives [6] and Anderson and Krathwohl's more recent revision of Bloom's taxonomy, AKT for short. AKT aims to accommodate new insights in cognitive psychology, curriculum and instructional design, and assessment. Both taxonomies describe six levels of cognitive performance with increasing complexity. In AKT, they are labeled: "Remember", "understand", "apply", "analyze", "evaluate", and "create". "Remembering" requires students to recognize relevant knowledge or recall it from long-term memory, while being able to "create" refers to the ability to devise a plan, put building blocks together to form a coherent or functional whole, to reorganize components into a new structure, or produce new artifacts.

In this section, we use this taxonomy to qualify and relate learning tasks and activities and combine them with different instances of information. We call this *didactic parameterization* of information objects and illustrate its use for the topic area "finite automata". A finite automaton (or state machine) represents an abstract mathematical model of a physical of mental machine with a memory. Finite automata are frequently used as modeling tools in different disciplines, including computer science, engineering, linguistics, or biology. Even learning designs have been modelled with finite automata. A finite automaton can be represented as a mathematical structure, a visual state transition diagram (see Fig. 3), or a transition table. In addition, a finite automaton is a computing device that accepts a regular language.



Task: Find a sequence of input strings leading to an accept state.

Fig. 3. Learning object with state transition diagram as information object

Figure 3 shows a learning object including a state transition diagram and a learning task. In AKT, this learning object would range at the second lowest cognitive process level "understand". To this end, we assume that a student has studied the basics of finite automata and is about to test his or her learning achievements. What we expect from students to recall here is simply the concept of finite automata, their behavior in terms of inputs and state transitions and their relationship to regular languages. Students can enter their solution in the window at the bottom and their input is immediately checked based on standard algorithms.

Following the late composition principle, the learning object depicted in Fig. 3 is maintained as a relation rather than a closed object in the repository network underlying the portal Edu-Sharing. The relation consists of an information object (here: the representation of an automaton in the form of a state transition diagram) and the specific didactic context (here: the learning task description).

The flexibility of this approach derives from the fact that it allows us to combine a single information object with different didactic contexts that are organized along Anderson and Krathwohl's or any other suitable educational taxonomy. For instance, the automaton in Fig. 3 could have been used by other teachers in combination with the following learning tasks that address different cognitive levels:

1) **Remember**: Define the mathematical structure of the automaton shown in Fig. 3.
Determine whether the diagram denotes a graph, a tree, a Petri net, a communication protocol, or a finite automaton.
2) **Understand**: Provide a sequence of strings that leads the automaton in Fig. 3 to an accept state.
Determine whether the automaton in Fig. 3 will accept the following sequence of strings: card inserted, [card=valid], enter PIN, [PIN=valid], cancelled.
Develop a transition table that is equivalent to the automaton in Fig. 3.
3) **Apply**: Provide the regular language that is accepted as input by the automaton in Fig. 3.
4) **Analyze**: Assume that the automaton in Fig. 3 models the behavior of an automatic teller machine. Determine how many states and transitions need to be added to the automaton in Fig. 4 to model the case that a bankcard is withdrawn after three failed attempts to enter a PIN. Expand the model correspondingly.

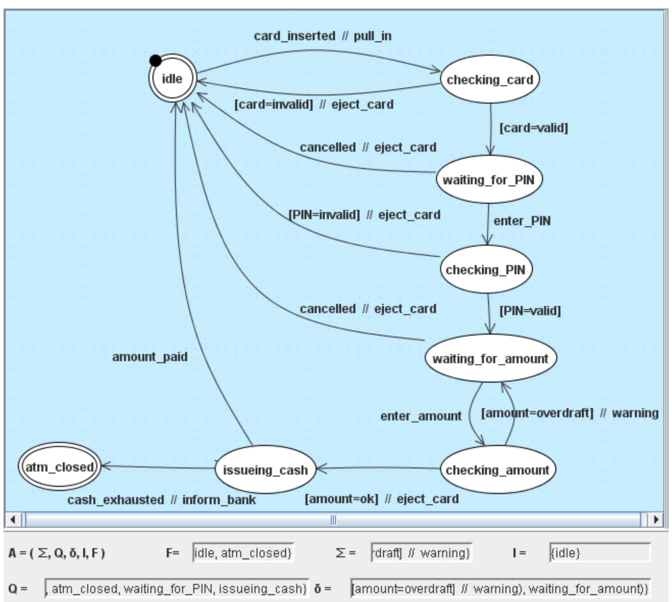


Fig. 4. Student view of the visual ATM model; the student has defined mathematical structure corresponding to the graph

A user who finds, e.g., the learning object shown in Fig. 4, will also be informed that this information object has been combined with other didactic contexts, which are listed above.

Conversely, these learning tasks could have been used together with another automaton that better fits into the larger context of another teacher's course. In Fig. 5, for instance, the learning task from our first example in Fig. 3 has been used in combination with a different information object. It models a simple bottle sorting machine for large and small bottles (lb, sb) that are, e.g., delivered via a conveyer belt and need to be sorted by removing bottles one by one from the belt and dropping them into a box for small or a box for large bottles, respectively (rsb, rlb).

Once an object like this is found, all combinations of this object in other contexts are listed to stimulate authors and re-users to provide parameterized objects and build on others' work.

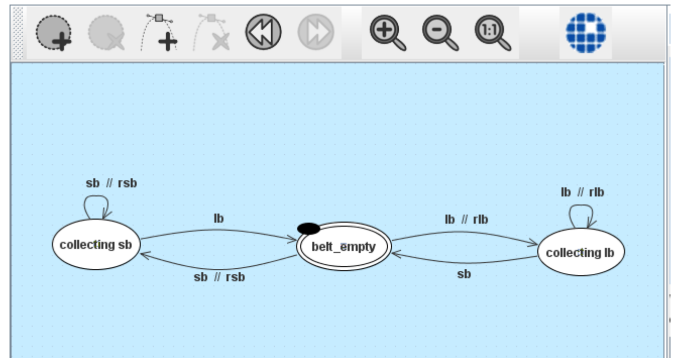


Fig. 5. Model of a bottle sorting machine reusing the didactic scenario from Fig. 3

The portal Edu-Sharing includes a Java-based editing tool that allows users to modify automata and create new ones (as state transition diagrams, transition tables or mathematical structures), to adapt an existing didactic context or define a new one, and recombine existing or new elements. Students can also execute a specified input sequence on a given automaton to determine whether their understanding of an automaton's behavior is correct. Figure 6 illustrates the preview an author can select to test the student's perspective before publishing his or her work. For automata-based learning objects associated with lower level cognitive tasks, the students' response can mostly be checked automatically based on the semantic equivalences between different representations of automata.

Of course, modifications to information objects and didactic context are only enabled if the re-user owns the right to do so (see also Section VIII).

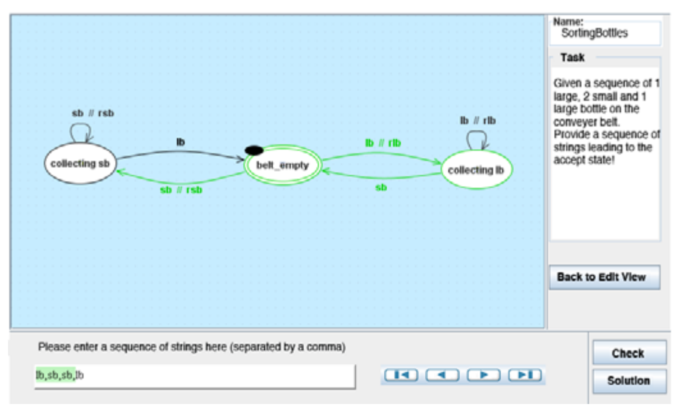


Fig. 6. Executing the model of a bottle-sorting machine in preview mode

We plan to build similar environments for graphs and, as special types of graph, trees and Petri nets. These objects share important properties with finite automata that support a didactic parameterization: They have a visual representation that can be used to model a rich set of real-world problems ranging from social network analysis over routing problems in street or communication networks to coloring problems. This bears the potential for defining higher-level cognitive tasks of type application, analysis, evaluation, and creation. They come with a well-defined mathematical theory and are related to other theories like linear algebra. Finally, the theoretical underpinning provides the basis for a rich set of traversal and manipulation algorithms, which can be exploited to test a student's response automatically.

IV. Design and Implementation of Adaptable Interactive Learning Objects

Adobe's Flash is a popular media type for implementing animated learning objects. Advantages include powerful animation and interaction capabilities, availability of Flash players and plug-ins on many operating systems, and ease of distribution and integration into interactive courseware. However, apart from didactic competence, the development of high-quality Flash animations requires know-how in media design and technical animation skills. Therefore, it will often be a better choice for a course author to reuse and – if necessary – adapt suitable animations from others rather than developing them from scratch.

In this section, we will illustrate the design of customizable Flash animations with two examples of learning objects that previously proved to be useful in higher education. The advantages and drawbacks of the two methods employed will be discussed in detail. We also describe the architecture of the technology used, which relies on Adobe's Flex framework.

A. Adobe Flex Framework

Flex is a new technology proposed by Adobe. It aims at providing a free, open source framework for building highly interactive Web applications. Flex applications are compiled into Flash (.swf) files that can be deployed and run consistently under major browsers and operating systems. The Flex framework provides a standards-based language and a programming model that supports common program components, in which user interface (UI) design and client logic implementation are clearly separated. MXML, a declarative XML-based language, is used to describe UI layout and behaviors. ActionScript 3, a powerful object-oriented programming language, is used to create client logic. These features of the Flex framework provide several possibilities to develop reusable animated and interactive learning objects.

In Flex, Flash animations can be generated by compiling the MXML text file, which may represent the template for a family of animated objects. If properly parameterized, each template can be configured differently by different instructors to accommodate their individual didactic context. The configured template can then be compiled into different versions of the generic Flash animation. As the Flex framework takes a component-based programming paradigm, a Flash animation itself can become a programmable object. This provides the basis for developing information objects that are largely free from context and expose possible animations through a programmable interface. The re-user then only needs to take care of the desired didactically meaningful interaction behavior. Figure 7 depicts the conceptual architecture of the proposed method. Concrete application examples are presented in the following two subsections.

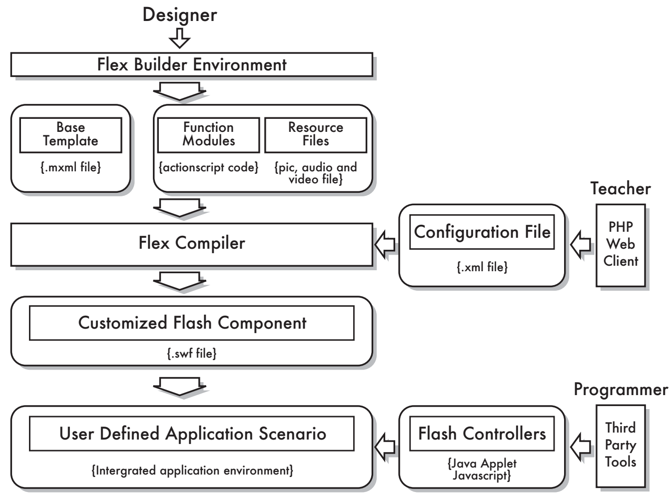


Fig. 7. Conceptual architecture of Flex-based learning object design and customization

B. From Object to Template

Besides the advent of Flex, a motivational element behind our approach was the desire to reuse a simple interactive Flash animation in a different thematic context without the need for editing the Flash file. A simple example is shown in Fig. 8. This animation aims to test the following educational objective, which would reside on the comprehension level of Bloom's or Anderson and Krathwohl's cognitive taxonomies:

> Given a set of concepts that were raised prior in this course in a case study illustrated by a number of authentic car rental scenarios, the student in a beginner course on object-oriented programming will be able to accurately sort 20 concepts into the three categories 'object', 'attribute' and class' within less than two minutes.

Figure 8 depicts the students' view of an interactive Flash animation currently in action. When the animation is started, a list of the concept will begin to move down the screen from top to bottom and thereby increase in size. The student has the task to pick the terms one-by-one with the mouse cursor and drop them into one of the three folders. This activity continues until all concepts have been sorted properly or the student gives up. Concepts that were dropped in the wrong folder will reappear in the scroll-down list.

As conceptual knowledge is important in any scientific and technical field, we designed a configuration environment for building concept classification objects from a Flex template. In this redesign process, we also included further parameters to control the interaction such as a timer, an error counter and a scrolling speed parameter.

Fig. 8. Screenshot of a Flash animation for concept classification

C. Customizing Parameterized Animated Learning Objects

Obviously, the concept classification animation can be used in a range of subject areas and disciplines including biology, software engineering or physics. To reuse and adapt the original Flash animation, it is, however, necessary to have access to the source file, the right to modify it, a Flash authoring tool or IDE, and sufficient Flash skills to implement the desired changes. This would correspond to white-box reuse in software engineering, which is the core of open source developments.

In this section, we will illustrate how the generalization and customization of such a learning object can be achieved through thematic parameterization. First, we need to generalize the educational objective to make it independent from the concrete case:

> Given a set of sample concepts and definitions of subject-related concept categories, the student will be able to accurately sort these concept into a predefined number of categories in a predefined time or with no more than m false classifications.

Further we need to create a template that allows the teacher to name the n concept categories desired, n sets of concepts to be used as test cases, one for each category, and n icons visualizing these categories. To provide additional flexibility, we introduce a range of parameters for defining

- The number of errors allowed,
- the maximum amount of test time,
- the rolling speed,
- the explanatory text including hints how to use the animation,
- the educational objective,
- background color, font, minimal, maximal text sizes, and other visual attributes.

To indicate the number of errors made and the time used for the test, we also need an error counter for each category and a timer.

Figure 9 partly shows a configurable Flex template implementing these features. For pragmatic reasons like screen

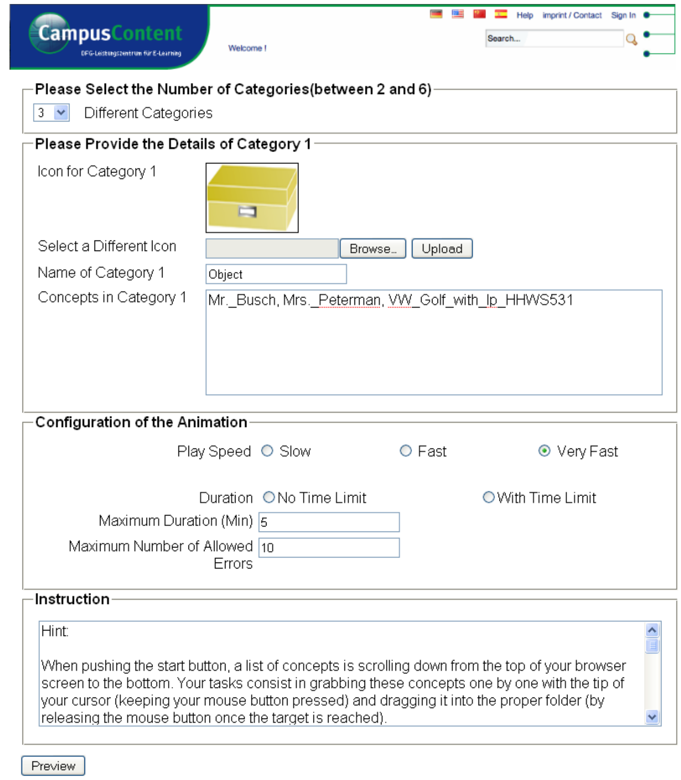


Fig. 9. Configuration interface of the parameterized animation

presentation and complexity of use, we allow between two and six different categories. The preferred icons representing concept categories can be uploaded from the teacher's computer and textual elements can be copied or typed into the text windows named "Concepts in Category i". Once all desired modifications are made, the re-user can activate the "preview" button to view the customized animation. The configuration data will be written into an .mxml file from which the server-based Flex builder will compile the new Flash animation, which is presented at the client side. The final Flash version can be downloaded or – in the case of Edu-Sharing – be stored in the portal's repository and referred to in different courses.

Figure 10 depicts a customized version of this template that is used in our course "Object-oriented Programming" in place of the original version shown in Fig. 8.

What Edu-Sharing users will find when searching the repository, are just fit-for-purpose objects like the one shown in Fig. 10. Compared to many other objects in the repository, the ones derived from a template carry a button "Customize" at the bottom, which suggests that such objects can be adapted. When clicking this button, the Flex-based template editor will be launched and the re-user can manipulate its parameters and produce animations that satisfy their needs.

D. Reuse of Animated Learning Objects as Software Components

The parameterization method discussed in the previous section provides a simple and effective way to customize animated learning objects without requiring special capabilities from the re-user. But it also exhibits limited flexibility because

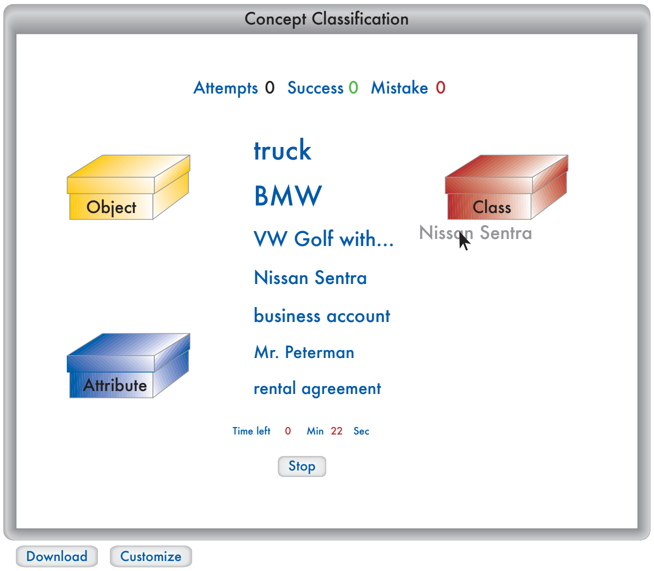


Fig. 10. Screenshot of configurable animation "Classify domain concepts"

the re-user has almost no possibility to change the didactic design. This is due to the fact that all the possible interactions between the flash animation and the user are hard-coded in the .mxml file of the base template. The teacher configuring it can change the appearance and set certain parameters limiting the animation behavior but he or she cannot modify the application logic.

To overcome this restriction, we propose a second reuse method. As already mentioned, the lowest layer of Fig. 7 suggests that the Flex framework allows each compiled flash animation to be used as a software component that may interoperate with other components. To further enhance an animation's adaptation capabilities, we propose to just define generic animation movements for the base template rather than a particular interaction sequence. In addition, a set of functions to activate such movements is exposed to the environment of use in the form of application programming interfaces (APIs). Re-users can use these APIs to define their preferred control sequences accommodating different didactic scenarios without the need to touch the Flash template or the resulting animation.

Fig. 11. Customizable "Bottle" Flash Animation

In Fig. 11, we show a Flash animation for a variable set of bottles. A simplified API for this animation is listed in Table I. Animation and API can, for instance, be used to visualize the behavior of the bottle sorting machine discussed in Section III and Figs. 5 and 6. We could write an algorithm that creates large and small bottles one by one and moves them from left to right from position 0 to 9 and then removes them again one by one. This behavior would simulate a conveyer belt. Then we could define that a small bottle has to be removed from position 6 and a big bottle from position 8 to simulate their sorting into different boxes. The transitions in Fig. 5 that are labelled with the input strings sb and lb (for small and large bottle, resp.) and have no output string could then be equated with a "detect bottle size and move right by one position" operation for all bottles left of position 6 or 8, respectively. The transitions labelled sb//rsb could be equated with operation removeBottle(6) and those labelled lb//rlb with operation removeBottle(8).

TABLE I
LIST OF THE KEY APIS

Operation	Intended meaning
setBottle(s,c,i)	Create new bottle of size s and color c and place it at position i
getBottleNumber()	Get number of bottles
isEmpty(i)	Test whether position i is empty
getSize(i)	Get size of bottle in at position i
moveBottle(i,j)	Move bottle at position i to position j
highlight(i)	Highlight color of bottle at pos. i
removeBottle(i)	Remove bottle at position i

In another context, we could use the "Bottle" animation in combination with a sorting algorithm controller that allows us to apply different sorting algorithms to an unordered collection of bottles of different size. Students could be asked to observe a sorting animation and determine the actual algorithm that was applied and reason about their insights.

To give another example of the advantages of decoupling visual representation and animation control, Fig. 12 shows a combination of a map of Germany and a controller implementing different graph traversal algorithms including breadth-first, depth-first, and Dijkstra's algorithm. The map shows connections between major cities, which represent the nodes of the graph, while connections are visualized as edges.

A learning task could then be to determine the shortest route between two cities A and B, where the distance is determined be the number of edges between A and B. Alternatively, the edges could be labelled with kilometers or another metric and the task would be to determine the cheapest connection between A and B. The screenshot of the map shows a situation in which Dijkstra's algorithm is used to measure the distance between Hamburg and Munich.

Each connection between two cities can be highlighted throughout the animation using the APIs of the animation. Through these APIs, a third-party program can also query the weight or distance associated with each connection and the currently selected node(s).

The lower part of Fig. 12 illustrates how a third-party program can make use of this animation. This behavior can be used to

- visualize an algorithm's behavior in the form of changes on the map,
- let a student control the manual execution of the al-

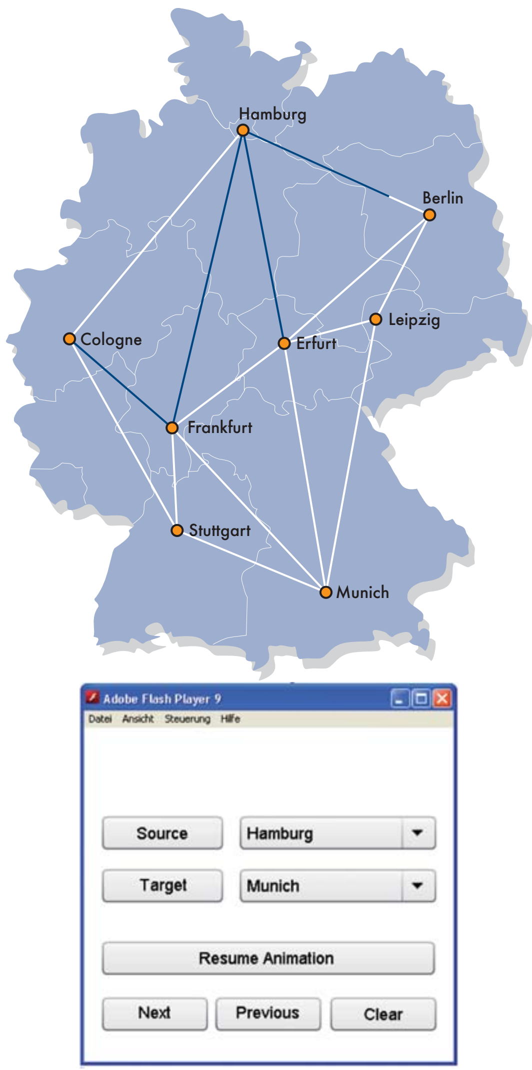


Fig. 12. Interacting with different graph algorithms

gorithm by clicking on selected edges in the proper sequence, or

- implement other learning tasks.

A third-party program can control the whole animation process through the APIs provided by the graph animation.

V. Animated Learning Objects Viewed as Software Components

In the previous section, we discussed two different approaches for developing reusable interactive learning objects. To enhance their reuse potential, we suggested a further separation of concerns. We proposed to define the visual appearance of an animation and an unconstrained behavior with the help of a Flash editor and implement meaningful behaviors in the form of controllers that are interfaced with each other through APIs. This approach has the desired side effect that both components can be maintained separately as long as the interface remains stable. Again, this is a design principle that has been exploited successfully in component-based software engineering.

As we pointed out in the introduction, learning object development is a complex process that involves different competencies such as instructional design, media design, programming, and domain expertise. It is unlikely that one person owns all these skills. Therefore, we believe that an effective reuse paradigm for learning objects should be leveraged to a higher degree of productivity by using the best fitting technology and flexibly organizing the cooperation of necessary competence holders. Based on his or her own expertise, a re-user can choose the corresponding level of customizing learning objects. To support such processes, we aim to provide a collaborative software environment in which re-users with different expertise can work together seamlessly.

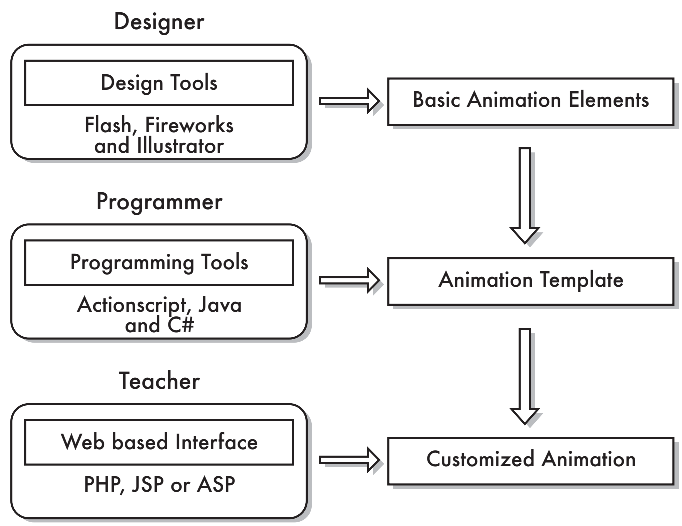


Fig. 13. Reference framework for scalable reuse of animated learning objects

In Fig. 13 we present a reference framework for the scalable reuse of animated learning objects described above. Within this framework, re-users at each level are supposed to work largely independently while at the same time being able to benefit from their mutual contributions. This framework can be realized with the FLEX environment and the possibilities that other tools provide. Java applets have been investigated to some extent as control components for Flex generated animations. In programming education, for example, this feature would allow students to implement their own animation control algorithm in Java.

While the framework sketched above is technically realizable, the tasks of re-users at each level are related to each other and cannot be separated so clearly. For example, to program the application logic, the developer must communicate with the instructor to understand the didactic scenario and requirements to be satisfied. Within a closed group, this may be easier to solve. However, when considering this issue in the context of an open collaborative platform, it will be difficult for re-users with different backgrounds to express their requirements and locate the appropriate resources. We aim to address this problem by defining a unified description schema that can be understood and used by all persons involved in such a collaborative design process.

VI. Design for Reuse and White-box Adaptation

To illustrate the advances that can be achieved through a continuous strive for reuse potential, we want to report on the evolution of modular learning materials for introductory statistics courses for different disciplines.

A closely collaborating colleague began with an initial version of a multimedia course that was built with a proprietary authoring tool and was delivered on a CD. The reuse potential was close to zero, even for the author himself. After spreading the message about reusable content, a new version was produced in HTML, including a large number of Java applets, audio and video files, and animations. In principle, reuse possibilities had increased, but only if the author was willing to provide the source code of his applets and other mulimedia components. The explanation is that none-HTML components in this course are just referenced from locations to which third parties have no access.

To overcome this weakness, the author now offers a collection of outstanding interactive learning objects in statistics [25] that can be downloaded from his home page [26] or referenced from within the Edu-Sharing repository. Many of these objects include textual and spoken explanations in German, English, French and Spanish, which suggest another mechanism to increase reuse potential: language parameterization. Due to the quality of this work, some objects have been translated into Japanese and are used at Japanese universities. This work is closely connected to similar work of other professors teaching statistics at other German universities. Their joint work towards the goal to develop the foundations of a new approach towards statistics education (New Statistics) was financially supported be the German Minister of Research, and the outcomes are currently used and maintained by 10 German universities.

Figure 14 shows a snapshop of an interactive experiment taken from [26]. It refers to the Gini coefficient or index that is a statistical measure to represent unbalanced distributions. It can be used to investigate and explain the important question of equal distribution of income, wealth, power and influence, or markets that are relevant in different disciplines, including business management, economics, or political sciences The situation of the experiment depicted in Fig. 14 indicates a deviation from equal distributed in the shaded area underneath the $45°$ line. This state could be the result of a student who tried to solve the following learning task:

> Interpret the Gini coefficient and demonstrate what it represents by modifying the sliders in the animation such that you obtain a deviation from the ideal Lorenz curve. Then identify the percentage of superstores that generates 50

Although this object is extremely well designed, a second glance reveals some potential for improvement towards a higher degree of reusability. Both paragraphs of the explanatory text (and audio) refer to a concrete example, which might not fit the context of another author's course or lecture so well. Others may want to add further recommended interactions and learning tasks. To enable this, they need to have the right from the author of this experiment to change its content, they need access to the source code and they need a proper tool to operate on this source. We call this *white-box adaptation* as opposed to the *black-box adaptation* we discussed in previous sections. Black-box adaptation only manipulates the interface of an object, while white-box adaptation modifies the object's interior and as such it ressembles the open source software development approach.

Edu-Sharing is open to all types of adaptation and reuse. Only content authors can impose constraints with the type of use license they declare and a lack of proper editing tools or skills on the re-user's side can prohibit white-box adaptation.

In the following section, we discuss a practice-oriented didactic model that allows us to cure the flaws of the Gini experiment by separating out those parts that are likely to be changed by re-users into the different facets of a learning path. However, before doing so, we summarize a few observations aiming at good design of information objects. Some of these principles are specific to the topic; others have been inspired by "design-for-reuse" principles in software engineering.

- Avoid verbal references to external sources;
- avoid hyperlinks to resources not accessible in the given virtual learning environment, here, the portal Edu-Sharing;
- find and isolate topics, concepts and notations of expected variability and try to use parameterization to handle contextual variability;
- constrain an information object to common invariant content;
- use aggregation and hierarchy to compose more complex objects from simpler ones;
- design information objects as if they were stand-alone products;
- capture context, educational aspects and documentation in proper facets of learning paths (see next section).

VII. Learning Paths

In an early phase of the CampusContent project, we studied the IMS Learning Design (LD) standard [15], [20] and a few prototype versions of LD editors intensively with the intent to use this standard as a basis for technology development. The experiments with these editors were, however, not encouraging because they exhibited too many usability weaknesses. However, more importantly, in many conversations with potential Edu-Sharing users, we recognized that the LD standard is not yet popular in educational practice. As LD will be better received, a usable LD editor will be included in the portal Edu-Sharing.

For now, we decided to adopt a pragmatic approach to codify learning processes in the form of learning paths. A *learning path* is a sequence of learning phases an individual follows to acquire knowledge, skills and competences. This approach builds on many of the concepts promoted in LD but presents them in light version. Our phases just distinguish two roles, teacher and learner, while LD allows the definition of arbitrary many roles. A phase involves a learning objective, resources and activities. Activities can be supported by e-learning, cooperation, and communication tools. Each phase

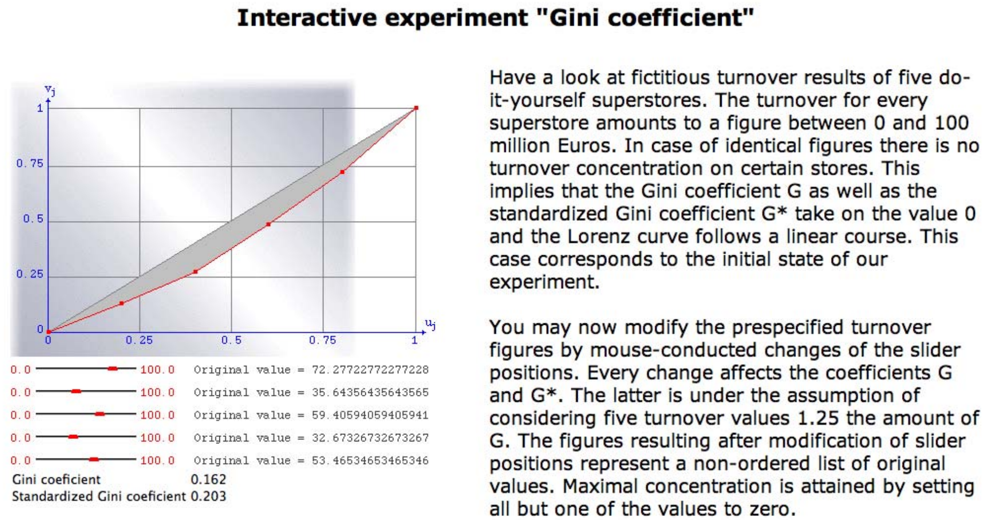


Fig. 14. Interactive experiment to deepen the grasp of the concepts Lorenz curve and Gini coefficient

may include information or learning objects that are thereby aggregated to higher levels of granularity.

Figure 15 shows the first phase of a blended learning approach that is based on the well-known project-based didactic scenario. The complete scenario consists of seven consecutive phases:

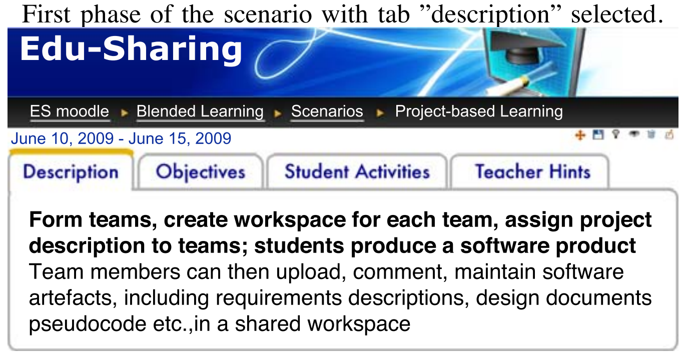
1) Team formation and initial setup (the phase depicted in Fig. 15);
2) requirements acquisition and evaluation;
3) draft design;
4) comparison and evaluation of different team solutions, refinement of preferred design solution;
5) implementation and testing;
6) test evaluation;
7) archival of project results.

The first three phases are organized as self-study phases for geographically dispersed students who communicate and interact with each other and with the tutors using the Edu-Sharing's workspaces, email, a wiki, and a forum. Phases 4 and 5 are organized as face-to-face meetings in a location that provides access to professional software engineering tools. Phases 6 and 7 are again self-study phases. Earlier versions of this scenario have been used (with other means) several times by the first author to conclude a two-semester distance-learning course on software engineering.

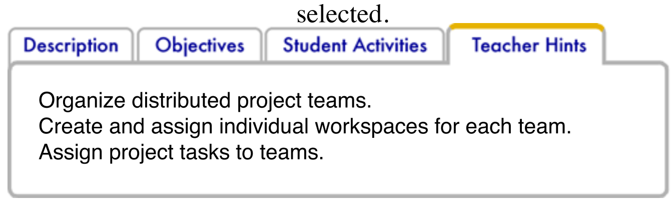
The icons in the lower right part of the "Student Activities" pane indicate that the students' activities are supported by a wiki, a forum and a document folder. The teacher who adapted this scenario to her or his needs has specified this. Resources comprise learning objects, learning units, and arbitrary types of documents, while activities include individual and group activities, interaction and communication activities. The different tabs may include links to online material stored in the repository, an Edu-Sharing workspace, or elsewhere on the Web. It can also specify offline resources to look at in this phase.

Reusable scenarios should be independent of a particular

First phase of the scenario with tab "description" selected.



First phase of the scenario with tab "hints for teachers" selected.



First phase of the scenario with tab "recommended student activities" selected.

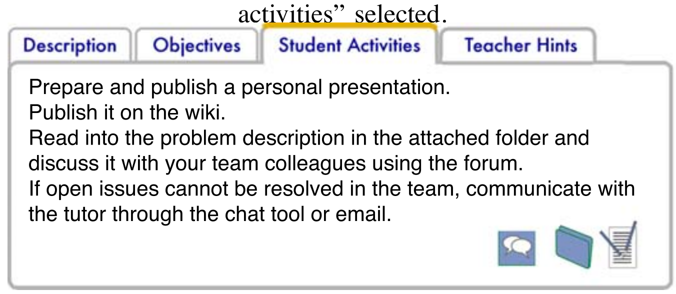


Fig. 15. Different sections of the top part of a blended learning scenario maintained in Edu-Sharing and presented in Moodle

discipline. Edu-Sharing offers a growing number of mature scenarios of different granularity that we adapted from literature and codified using the concepts discussed in the previous paragraph. Fine-grained examples include devils advocate, active structuring, flashlight, brainstorming, concept mapping, think-pair-square and webquest. More complex scenarion that typically rely on tool support include case study, jigsaw classroom, puzzle method, strategic problem solving and others. They are published in the form of generic templates, i.e., without specific resources and tools, in the portal. To facilitate search and finding, these templates are supplemented with appropriate metadata, which have been defined by the project CampusContent (LOM and Dublin Core are metadata standards offered to decorate information objects).

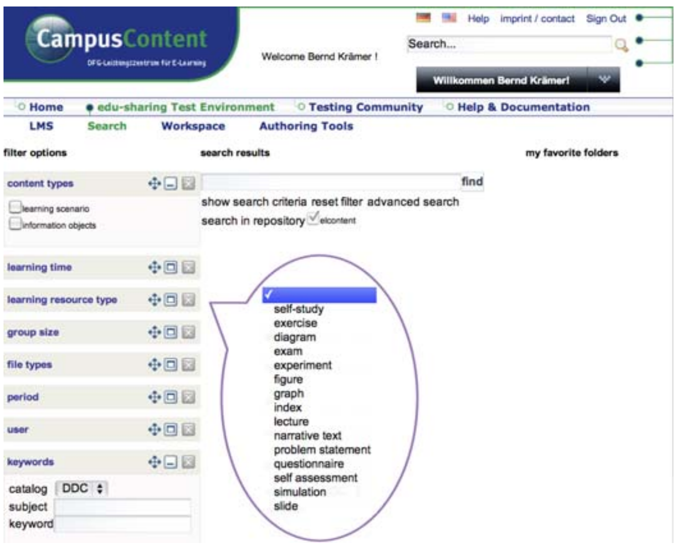


Fig. 16. Searching for content in the repository

Figure reffig:search shows a few filter options that Edu-Sharing users can select in the process of searching information, learning objects or scenarios. The window that pops up when selecting the filter ”learning resource type” is shown in the ballon in the center pane. Re-users who find such scenarios and want to adapt them to their needs, can edit them with the help of Edu-Sharing's scenario editor. We expect that such scenarios may inspire educators, who had no clear idea before of what a didactic scenario is, to try them out in their own teaching.

VIII. Open Content, Intellectual Property Rights and Licensing

An open environment for exchanging intellectual property requires clear and legally well-defined regulations to ensure that the interests of both rights owners, i.e., the authors of information and learning objects and of didactic scenarios, and re-users like trainers, teachers, lectures etc. are respected and served. Content authors want to maintain their right to decide what others are allowed to do with their intellectual property. Potential re-users need the certainty of the law and more than just trust in the availability of third-party content. It is also in the interest of portal operators to limit liability to their sphere of influence.

”Open content” initiatives and their specialization ”open educational resources” have been inspired by the open source software (OSS) movement that promotes licenses allowing the free access to source code and its unlimited non-commercial distribution, reuse and adaptation. In the late 1990s Wiley and others adapted this definition to digital content of various types including text, image, graphics, audio, video, animation and the like. In contrast to the ”all rights reserved” claim of the classical copyright, open content requires subtly differentiated possibilities for organizing use and modification rights. This is addressed by a range of dedicated license models including the GNU Free Documentation License [12], originally designed for documenting OSS, the OpenContent License [30] or the more recent and relatively popular Creative Commons License [10].

By default, the portal Edu-Sharing supports Creative Commons but other license models can be made available as needs of certain user groups arise. A license manager pops up whenever new content is uploaded in the portal's repository network. It allows content owners to select a proper license and informs content users about the license conditions of particular content objects found. In addition, when composing several information or learning objects to a larger units of learning, the license manager detects incompatible licenses imposed on compound objects.

IX. Conclusion and Future Work

Already in his early seminal paper from May 1975 entitled *Guidelines for a general didactic concept for the development of study materials in distance education*[3], Otto Peters [29] stressed the need for adapting didactic elements like objective, topic, method, and media to learning situations found in distance learning. This leads to the intertwining of learning objectives with instructional methods and media, whereby the media have to be tailored to the actual setting. The main functions of technical media include content representation, contact medium, and illustration and visualization material, which should be systematically evolved in media didactics that is concerned with the planned, targeted and reflected use of technical media for educational objectives and purposes. The concept of learning objects promoted by CampusContent conforms to Peters' idea of intertwined facets consisting of information, a learning objective, and a didactic scenario that describes what the object can be used for and how learners will interact with it. To provide a high degree of flexibility, we allow re-users to dissect a learning object and recombine it differently.

This article particularly focused on methods and a scalable framework for developing and personalizing customizable interactive learning objects. Three case studies based on real applications have been presented. The technologies used include Java and the Adobe Flex environment.

[3]Title translated by the authors. The original paper is written in German.

The core motivation behind this work is a contribution to the realization of a knowledge building, sharing and improvement cycle (see Fig. 17), which was inspired by [8] and in which:

- educational content authors are supported effectively in the process of creating added-value adaptable information objects and representation of best didactic practices in the form of scenario templates that can be flexibly associated with information objects and learning objectives;
- lecturers and teachers are encouraged to review and analyze the educational knowledge captured in learning objects and thus learn from the knowledge of their peers;
- lecturers and teachers are enabled to adapt and integrate the knowledge of peers and integrate it in their own knowledge.

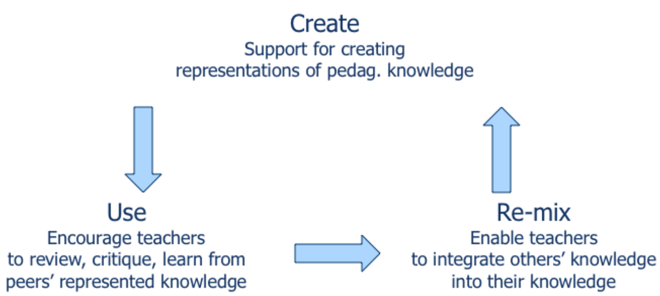


Fig. 17. Educational knowledge building and sharing cycle

In addition, we presented a technical infrastructure and portal, Edu-Sharing, that supports the sharing and reuse of learning objects and mature didactic scenarios. The infrastructure includes a network of repositories for sustainable storage and effective finding of reusable content, a range of tools for didactic scenario and content authoring, a licence manager, and community support including personal workspaces that can be shared with trusted peers. The portal can be easily interfaced with existing learning management systems (LMSes). The open source LMSes Moodle and metacoon will be included in the standard distribution. Others like Olat[4] or Ilias[5] will be supported soon. Open interfaces also allow the adaption of external user management systems and external data stores maintained by commercial providers, such as schoolbook publishers. The license manager will control accesses to such external stores based on access rights defined in local or foreign user management systems. For instance, a school may maintain information about user rights to access learning materials from Klett International. This information is used transparently by the license management component of Edu-Sharing to route an access to object O of a student from that school to Klett's database if the student's teacher owned and passed the right to access O to her students.

Currently, the portal Edu-Sharing undergoes a pilot test with different user groups including university lecturers, high-school teachers, and vocational school teachers to evaluate different use scenarios and identify bugs and usability flaws in the software. A revised version of the portal software will go public by the end of 2009. Therefore, we are currently lacking sufficiently large user groups to provide more mature evaluation results.

The anticipated added value of the project resides in the benefit lecturers and course authors gain from relying on previous work of their colleagues in subject areas bearing similarities in content and instructional design. As a result, users will have at their disposal an additional capacity for the improvement in specific areas of teaching. However, to achieve this goal, a critical mass of content and active participants in every subject is necessary. Therefore, networking of institutions and community building are currently major tasks of the CampusContent project management, besides supervising the pilot application phase.

Critics may argue that there is not much rich content available in (higher) education. This is even true in view of the open educational resources movement, which evolves into a world-wide community effort that includes milestones such as MIT's Open Courseware initiative or, more recently, the Open University's OpenLearn initiative and others. In addition, the coarse granularity of these resources limits their potential for reuse in other contexts. But this is presumably just a contemporary observation. In an interview with Richard Katz Andy Lane, one of the key figures behind OpenLearn, stated [19]: "...we shift from delivering relatively static content embedded in books and printed materials to delivering dynamic content via the Internet", and a bit further down the lane he said: "We are investing in more multimedia content, more simulations, more animations and video ...". Therefore, there is hope that the situation will improve as the symptoms are obvious.

References

[1] Alfresco: Open Source Enterprise Content Management System. Retrieved September 27, 2009, from http://www.alfresco.com/

[2] L. W. Anderson and D. R. Krathwohl. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon 2000

[3] P. Baumgartner. ROI: Reusable Object and Instruction Paradox. Keynote at the 9th European Conference on Media in Higher Education (GMW), Graz, Austria, 2004

[4] P. Baumgartner and S. Heyer. The Trinity: Towards a Model of Learning Objects. Draft paper, 2009, retrieved September 27, 2009, from http://www.google.de/url?sa=t&source=web&ct=res&cd=8&url=http%3A%2F%2Fwww.peter.baumgartner.name%2Farticle-de%2Ftrinity_model_of_lo.pdf%2F&ei=bzG2Sq_DNdqI_AbEmpWoDQ&rct=j&q=baumgartner%2C+trinity&usg=AFQjCNHgIJt6RBgxuzqrSn4vz5NHLZAQ-g

[5] P. Baumgartner and M. Kalz. Wiederverwendung von Lernobjekten aus didaktischer Sicht. In *Auf zu neuen Ufern! E-Learning heute und morgen*, 97-107, Medien in der Wissenschaft, Vol. 34, Waxmann, 2005 (in German)

[6] B.S. Bloom. *Taxonomy of educational objectives, Handbook 1: Cognitive domain*. New York: Longmans Green 1956

[7] T. Boyle. Design Principles for Authoring Dynamic, Reusable Learning Objects. *Australian Journal of Educational Technology*, 19(1):46-58, 2003

[8] J.S. Brown and R.P. Adler. Minds on Fire? Open Education, *EDUCAUSE review*, 17-32, Jan/Feb 2008

[9] B. Collis and A. Strijker. Technology and human issues in reusing learning, *Journal of Interactive Media in Education*, vol. 4, pp. 1-32, 2004

[10] Home page. Retrieved September 27, 2009, from http://creativecommons.org/

[11] E. Gamma, R. Helm, R. E. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994

[12] Free Software Foundation (Ed.), *GNU Free Documentation License*. Retrieved September 27, 2009, from http://www.gnu.org/copyleft/fdl.html

[4] https://www.olat.uzh.ch/
[5] http://www.ilias.de/

[13] P. Han and B.J. Krämer. Generating Interactive Learning Objects from Configurable Samples. In: Proceedings of the International Conference on Mobile, Hybrid, and On-line Learning (eL&mL), IEEE Computer Society, 2009.

[14] S. Heyer. Pedagogical enrichment of information objects. *Journal of Integrated Design and Process Science*, 10(3):33-45, 2006.

[15] IMS Global Learning Consortium. Learning Design Specification. Retrieved September 29, 2009, from http://www.imsglobal.org/learningdesign/

[16] IMS Global Learning Consortium. Question and Test Interoperability Overview. Retrieved September 27, 2009, from http://www.imsglobal.org/question/qti_v2p0/imsqti_oviewv2p0.html

[17] IEEE. Draft Standard for Learning Object Metadata. Retrieved September 27, 2009, from http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[18] R. Jones and T. Boyle. Learning Object Patterns. *Interdisciplinary Journal of Knowledge and Learning Objects*, Vol. 3, 2007

[19] R.N. Katz. Open to Change: An Interview with Leaders of the Open University, *EDUCAUSE review*, 58-64, Mar/Apr 2008

[20] R. Koper and C. Tattersall (Eds.). *Learning Design: A handbook on Modelling and Delivering Networked Education and Training*. Springer 2005

[21] B. J. Krämer and H.-W. Schmidt. Components and Tools for On-line Education. *European Journal of Education*, 36(2):195-222, 2001

[22] B. J. Krämer and A. Zobel. Rollout of CampusContent – DFG-Competence Centre for elearning, *e-learning & education*, July 2008. Retrieved September 27, 2009, from http://eleed.campussource.de/archive/4/1417/

[23] Liferay: Enterprise Open Source Portal. Retrieved September 27, 2009, from http://www.liferay.com/web/guest/home

[24] metacoon: Learning Management System and Authoring Tools. Retrieved September 27, 2009, from http://www.campussource.de/org/software/metacoon/

[25] H.-J. Mittag. Virtual learning environments for statistics education and applications for official statistics. Biannual Meeting of the Korean Statistical Society, Seoul, 5 6 November 2004

[26] H.-J. Mittag. New Statitics: Interactive experiments. Retrieved September 30, 2009, from http://www.fernuni-hagen.de/newstatistics/applets/appletIndex_en.htm

[27] Moodle: Course management System. Retrieved September 27, 2009, from http://moodle.org/

[28] OECD, *Giving Knowledge for Free. The Emergence of Open Educational Resources*. Retrieved September 27, 2009, from http://213.253.134.43/oecd/pdfs/browseit/9607041E.pdf

[29] O. Peters, Hinweise zu einem allgemeinen didaktischen Konzept für die Entwicklung von Studienmaterialien im Fernstudium, *ZIFF-Papier 1* 1975 (in German). Retrieved September 27, 2009, from http://www.fernuni-hagen.de/ZIFF/ZP_001.pdf

[30] D.A. Wiley (Ed.). *Open Publication License*, 1999. Retrieved September 27, 2009, from http://opencontent.org/openpub/

[31] D.A. Wiley. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In Wiley, D. A. (Ed.), The Instructional Use of Learning Objects: Online Version. Retrieved September 27, 2009, from http://reusability.org/read/chapters/wiley.doc

Understanding Object-Relational Mapping: A Framework Based Approach

Christopher Ireland, David Bowers, Michael Newton, and Kevin Waugh
Department of Computing
The Open University
Milton Keynes, UK
e-mail: cji26@student.open.ac.uk, (D.S.Bowers, M.A.Newton, K.G.Waugh)@open.ac.uk

***Abstract* - Object and relational technologies are grounded in different paradigms. Each technology mandates that those who use it take a particular view of a universe of discourse. Incompatibilities between these views manifest as problems of an object-relational impedance mismatch. In a previous paper we proposed a new conceptual framework for the problem space of object-relational impedance mismatch and consequently distinguished four kinds of impedance mismatch. Here we show how that framework provides a mechanism to explore issues of fidelity, integrity and completeness in the design and implementation of an existing object-relational mapping strategy. We propose a four-stage process for understanding a strategy. Using our process we show how our framework helps to identify new issues, understand cause and effect, and provide a means to address issues at the most appropriate level of abstraction. Our conclusions reflect on the use of both the framework and the process. The information arising from the use of our framework will benefit standards bodies, tool vendors, designers and programmers, as it will allow them to address problems of an object-relational impedance mismatch in the most appropriate way.**

Keywords: Object-Relational; Impedance Mismatch; ORM; Framework

I. Introduction

In [1] we provide a new framework for understanding the problem space of object-relational impedance mismatch. If we address the root cause of an object-relational impedance mismatch problem rather than the symptoms as we do today, we will reduce the cost of software development by avoiding the quagmire described by Neward [2] and discourage others (such as [3]) from reinventing solutions.

A paradigm is a particular way of viewing a universe of discourse. Each paradigm comes with its own particular abstractions, organising principles and prejudices. There are a number of different paradigms in computing. Each paradigm has influence on both the process and artefacts of software design and development.

The combination of technologies based on different paradigms presents a set of problems for those responsible for the design and implementation of an application. We refer to each such problem as an impedance mismatch problem. People are inventive and proponents of one paradigm may believe that they have solved an impedance mismatch problem. Such a solution will typically involve using a subset of concepts from one paradigm to represent a concept in the other. It then becomes received wisdom within a community that there is a solution to a problem and that all those concerned understand the solution.

The relational paradigm has proven popular in the development of databases whilst at the same time the object paradigm has underpinned a number of programming languages and software development methods. The popularity of technologies that embody different paradigms in these two separate but essential aspects of software development means that inevitably they will be used together. Differences in abstraction, focus, language etc. between paradigms however leads to problems when these technologies are combined in a single application.

An object-relational application combines artefacts from both object and relational paradigms. Essentially an object-relational application is one in which a program written using an object-oriented language uses a relational database for storage and retrieval. A programmer must address object-relational impedance mismatch ("impedance mismatch") problems during the production of an object-relational application. For some authors [4] however there is no impedance mismatch. This is true for those developing an entire application using a single programming language such as Visual Basic, C++, Java or SQL-92[1] ("SQL") because each language is based on a single paradigm. Those who have to combine object and relational technologies and must work across paradigms have a different experience [5], [2]. The received wisdom is that these impedance mismatch problems are both well understood and resolved by current solutions based on SQL. For each such impedance mismatch problem however there is a choice of solution. We refer to each such solution as an Object-Relational Mapping (ORM). Each ORM strategy addresses problems of an impedance mismatch in a different way. We seek to understand the most appropriate way to address a problem.

During the development of an object-relational application based on SQL-92, the resolution of impedance mismatch problems involves many different roles and takes time and effort to achieve [2]. Neward [2] labelled the problem of impedance mismatch "the Vietnam of Computer Science" because initial quick wins based on the received wisdom are rapidly replaced by a quagmire of

[1] This work is presented in the context of mapping from an OOPL to SQL-92, which does not include Object Relational (OR) extensions. Future work will analyse the effectiveness of the OR extensions to SQL in addressing ORIM problems.

issues. Keller in [6] claims that twenty five to fifty percent of object-relational application code is concerned with problems of an impedance mismatch. The popularity of object and relational technologies, the plethora of solutions and technologies for the resolution of an impedance mismatch, and the existence of guidelines [7] and metrics [8] for selecting a strategy also suggest that problems of an impedance mismatch are neither uncommon nor trivial.

In this paper we propose a four-stage process for understanding a strategy. Using our process we show how our framework helps to identify new issues with a strategy, understand cause and effect, and provide a means to address those issues at the most appropriate level of abstraction.

The rest of the paper is structured as follows. Section II presents the impedance mismatch problem space; Section III presents an analysis of current approaches to ORM; Section IV presents our framework; Section V relates our framework to ORM strategies; Section VI presents our process for using the framework; Section VII provides a worked example; and Section VIII presents our conclusions and future work.

II. Problems of an Object-Relational Impedance Mismatch

In the context of object-relational application development, one objective of an ORM strategy is to isolate a programmer using an object-oriented programming language (OOPL) from the need to understand the SQL language, the schema of an SQL database, and its implied semantics. A programmer need not focus on how to store an object but on what to store and when to store it, and what to retrieve and when to retrieve it.

Such a strategy is typical of ORM products such as Hibernate [9] and Oracle TopLink. They provide a programmer with a virtual object database, presenting data in a relational database as if it were a collection of objects. However, ORM does not isolate a relational database from an object-oriented program. The design of a relational database must address issues such as data redundancy, data integrity, data volumes, access control, concurrency, performance and auditing. Impedance mismatch problems occur when these requirements are at odds with the design of an object-oriented program. These problems have been described by writers such as [2] and [5]. In Table I we have catalogued the issues emerging from their work as problems of an object-relational impedance mismatch (ORIM).

TABLE I. Problems of an Object-Relational Impedance Mismatch

Problem	Description of the problem and typical questions raised
Structure	A class has both an arbitrary structure and an arbitrary semantics defined through methods. A class may also be part of a class hierarchy. SQL-92 does not provide an analogy for a class hierarchy or support repeating groups within a column. How then do we best represent the structure of a class using SQL?
Instance	To conform to relational theory, a row is a statement of truth about some universe of discourse, but an object is an instance of a class and may have an arbitrary structure. How does a row correspond to an object and where is the canonical copy of state located? Essentially, how much of an object must we maintain in a database?
Encapsulation	The state of an object is accessed via methods. The state of a row has no such protection and may be modified by other applications. How do we ensure consistency of state between an object and a row?
Identity	An object has an identity independent of its state. This in-memory identity will be different between two executions of a program. Within the same execution, two objects with the same state are different if they have a different identity. The primary key of a row is part of the state of that row. How do we uniquely identify a collection of data values across both object and relational representations?
Processing Model	An object model is a network of interacting discrete objects and access is based on navigation. The relational model is declarative and access is set-based. The object and relational models represent references in different directions. A transaction may not require all the data about an object. How do we represent in, maintain in, and retrieve from a database a sufficient set of in-memory objects?
Ownership	A class model is owned by a programming team, a relational schema is ultimately owned by a database team, it may hold legacy data and may also be used by other applications. When things change how do we maintain the necessary correspondence between a class model and a database schema?

III. Object-Relational Mapping

In the literature and in practice we find many examples of ORM ([3],[10],[11],[12],[13], and [14]). Essentially an ORM strategy is how we address each problem of an impedance mismatch but in research and practice the term ORM is used to refer to a number of different things: for Fussell [11] it is a transformation process; for others ORM is something defined in the configuration of a mapping tool such as Hibernate [9]; whilst to others it is a pattern [13] or canonical mapping [14] used as the basis for a design transformation.

Practitioners recognise ORM as both a process and a mechanism ([5], p225) by which an impedance mismatch is addressed. As a process, ORM is the act of determining how objects and their relationships are made persistent in a relational database: in essence the selection of one or more patterns [13]. These patterns are based on the assumption that an object model is used as the basis for a database schema and that schema confirms to SQL-92. They do not help with the development of an object-based application that uses a legacy relational database.

As a mechanism, ORM forms the definition of correspondence necessary for the successful implementation of a particular pattern as one or more mappings. A mapping relates two representations in different implementation languages. In order to address an impedance mismatch, this mapping is codified as one or more transformations within some part of an application. For the developer of an object-oriented application that must use a relational database for persistence ORM may be all of these, impedance mismatch is also a fact of life [13]. We observe from this variety that there is no consensus on a single strategy for ORM and by implication how we address impedance mismatch. Each strategy addresses a different aspect of impedance mismatch. Some strategies focus on equivalence between a class and a table [13] (what to map) whilst others propose a unified query language [15] (how to map) or software architecture [11] (where to map). It is not clear whether any of these strategies address the root case of an impedance mismatch problem or whether they make do with the facilities available. Evidently when these writers use the term "impedance mismatch" they are not talking about the same thing. We require some form of organising principle which goes beyond received wisdom to facilitate an understanding and comparison of strategies, and which also recognises an essential aspect of the problem: the different levels of abstraction.

It is evident from Table I that impedance mismatch is not a single, well-defined problem. The different interpretations of ORM also indicate there is no single, well-defined solution. If we are to understand impedance mismatch we must understand the nature of these different problems and how they are addressed by different approaches to ORM. At a detailed level this understanding provides the motivation for our conceptual framework and classification.

IV. A Conceptual Framework of Object-Relational Mapping

In this section we consider how we might organise the different views of ORM. Other models such as [11] and [16] focus on client/server software architecture. Essentially they help inform where one might perform a mapping. Hohenstein [12] considers programming language issues and helps to inform a C++ programmer how to perform a mapping.

The rationale and the motivation for our framework and classification were established in [1]. Our framework comprises four levels of abstraction common to both object and relational technologies. The classification allows us to organise the different issues at each level. These levels (Table II) allow us to understand why we are performing a mapping and allow us to identify the root cause of a problem. Object and relational silos span all four levels. Within each level there are therefore both object and relational contributions. We summarise our framework in Figure 1.

The levels are labelled using terms that may themselves have alternative interpretations and therefore require clarification. A paradigm is one particular way of viewing a universe of discourse ([17], pA-6). A language is used to produce an abstract description of a universe of discourse. We consider a concept to be some identifiable collection of things from a universe of discourse. A schema is a description (representation) of some concept from a universe of discourse, expressed using a particular language. We consider program source code the schema for an executing program just as an SQL script is the schema for a relational database. Finally an instance is data about some thing from the universe of discourse set within a particular schema.

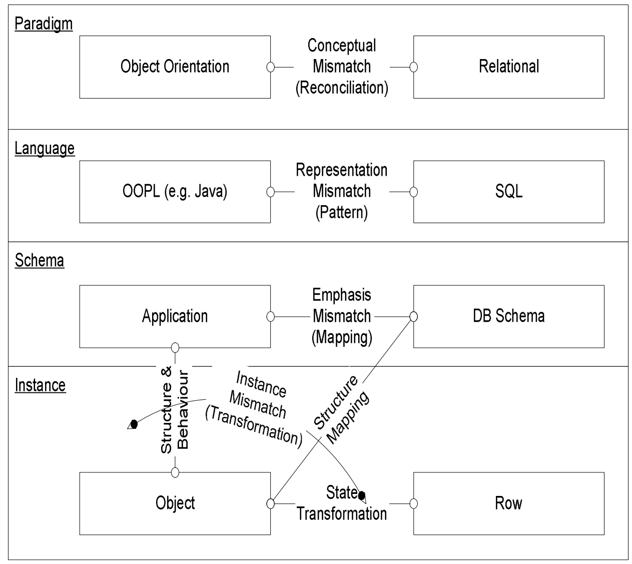


Figure 1. Our conceptual framework

The relationship between the levels of our conceptual framework is one of context. A paradigm sets the context for the semantics of a language. A language provides data and processing structures for describing the semantics of a universe of discourse in the form of a schema. There are many possible schemata. A schema sets the structure into which data about some thing from a universe of discourse must fit. Conversely a schema sets constraints on what it is we can represent about some thing from a universe of discourse.

TABLE II. Our conceptual framework of Object-Relational Mapping

Level	ORM is concerned with…
Paradigm	Issues relating to the incompatibilities between the two different views of a concept from a universe of discourse: one as a network of interacting objects and the other as a set of relations.
Language	Issues relating to the incompatibility of data structures between object and relational based languages. ([14], p182) refer to this as a canonical mapping. In this paper we will use the term pattern in the context of [13] as an outline description of a solution.
Schema	Issues relating to the maintenance of two representations of a particular concept described in different languages.
Instance	Issues relating to the storage and retrieval of an object in the context of an object-relational application.

Contextualisation has implications for choices made during the development of an object-relational application. A development language brings with it not only an implicit choice of paradigm but also a set of structures and patterns that may be used ([5], [12], and [13]). The maintenance of a legacy application may dictate the use of a particular language. Choices made during the design of an object model and an SQL schema dictate the content of a mapping schema. During the development of an object-relational application, the teams responsible for program and database schema development will make their own choices based on their own agenda.

A programmer has many technologies and algorithms from which she may choose in order to implement a transformation: for Java alone there is a choice of JDBC, Hibernate, JDO and Oracle TopLink to name a few. All levels of our conceptual framework have influence on the work a programmer must do in order to resolve an impedance mismatch. When we use the term ORM we must recognise that an impedance mismatch problem has its source at any of these levels, and understand how and at what level(s) a problem is best addressed.

Analysis of ORM strategies in the literature such as [13] have focussed on consequences in implementation rather than understanding the underlying issues with a strategy. Our framework provides an organising mechanism that allows us to explore issues in the design and implementation of existing and new ORM strategy choices. Achieving an understanding of the underlying issues is an important contribution of our framework.

V. ORM Strategies and our Four Level Framework

Our framework provides a new way to think about the problem of impedance mismatch and how we go about resolving it. Each level provides a different way of thinking about an ORM strategy. In this section we explore the relationship between problems of an impedance mismatch and the layers of our framework. For each strategy we provide illustrations from the literature.

A. Paradigm

An ORM strategy at the paradigm level involves the reconciliation of different perspectives of a universe provided by the object and relational paradigms. Different aspects of an object-relational application are grounded in each paradigm. Typically the object paradigm influences program design and the relational paradigm database design. ORM in this context is the act of bridging the differences between these two paradigms. This is the essence of the impedance mismatch problem. It is therefore important to understand the nature of these differences.

There is no consensus of terminology. Each paradigm uses a different set of building blocks to describe a universe of discourse. Although there is no single agreed definition of an object-based representation (the UML is one attempt but there are others [18]), such a representation will typically include concepts such as class, subclass, object, attribute, and association. There is however a single definition of what constitutes a relational representation [19]. A relational representation of the same universe will include concepts such as relation, tuple and domain.

There is some correspondence between the building blocks. The relational paradigm does not prescribe the domains that may be used. Neither does an object paradigm prescribe the objects that may be used. A relation represents an assertion (a predicate) about a universe of discourse involving one or more domains and a tuple of a relation is formally a statement of truth about that universe. There is however no equivalent representation in the object paradigm. An object is not a representation of a statement of truth about a universe of discourse. Furthermore an object has identity and encapsulates its state whereas a tuple does not. A class defines the allowable attributes and behaviour of an object but its definition is not based on predicate logic. Whereas the relational model is concerned with statements of truth, an object has arbitrary semantics. The behaviour of an object is defined using methods and a valid state is defined using a constraint. In this respect a tuple may be considered inert in so far as it has no intrinsic behaviour. Instead a tuple may be the target of a relational operator such as project, restrict or union. A lack of correspondence between two perspectives on a universe of discourse materialises as an impedance mismatch. We label this kind of impedance mismatch a *conceptual mismatch* and it is addressed using an ORM reconciliation strategy.

A reconciliation strategy must address differences in perspective, terminology and semantics. The designer of an object representation and the designer of a relational representation view and describe aspects of a universe of discourse in different ways. The designer of an object-relational application must identify correspondence and reconcile differences between these two perspectives.

One example of the reconciliation of object and relational semantics is provided by Date [20]. He emphasises that relational theory is not at odds with the ideas of object-orientation. Just as the semantics of a class are arbitrary, the relational model does not prescribe the data types that may be defined. There is therefore scope for addressing a conceptual mismatch.

B. Language

An ORM strategy at the language level is concerned with identifying general patterns of correspondence between the data structures available in an OOPL such as Java, and those structures available in SQL.

Each language reflects the paradigm on which it is based. The outline structure of a Java program is a collection of classes. A class may be viewed as a template for the creation of an object at run-time. An SQL schema is a description of a collection of tables. A table corresponds to a relation. Whereas, formally, a tuple is a statement of truth, the semantics of a row are less strict. A row represents data about some thing from a universe of discourse. Each row corresponds to a tuple.

A significant difference between Java and SQL is the extensibility of their type systems. Whereas a class is an essential part of the extensibility of the Java type system there is no equivalent extensibility in the SQL type system. Implementing a representation of a relation in an OOPL [21] or an SQL like syntax within an OOPL [15] may move the primary focus of ORM activities to the schema level, but it does not address this extensibility issue or remove the need for an ORM strategy.

Generally a class is part of the type system of an object-oriented program. Using SQL a class is something that may be represented, it is not an extension of the type system and also not a first-class citizen. This is the essence of a structure problem and there exist a number of patterns to help resolve this [13]. Aspects of an object-oriented design such as a class and an object fit into a *representation* that must be described using the existing features of SQL. A column is a scalar value and cannot adopt the type of a class represented in such a way. This representation is limited as it may only be used to store the state and not the behaviour of an object. In an object-oriented application at run-time an object has a unique identity independent of its state. The value of a primary key is part of the state of a row. This is the essence of the identity problem. The mismatch between two descriptions of a concept materialises as an impedance mismatch. We label this kind of impedance mismatch a *representation mismatch* and it is addressed using an ORM pattern strategy.

A pattern provides a way to describe correspondence between data structures. SQL provides an approximation of the data structures mandated by the relational paradigm, just as Java provides an approximation of those mandated by the object paradigm. The syntax and grammar for SQL is defined by standard and is implemented in vendor-specific languages such as Oracle and SQLServer. None of these languages is a pure implementation of SQL but nevertheless may be classified as a relational language. In practice we must address not only differences between languages as defined by their respective standards but also differences between vendor implementations. A pattern strategy must provide one or more patterns (such as [13]) that address issues of structure and identity.

C. Schema

An ORM strategy at the schema level will produce a mapping between two representations of a concept. Our emphasis here is on design issues. The description of a concept within an object-relational application will involve at least two schemas: one based on class and the other based on table. These two representations of a concept are different not just because they are phrased in a different language, but because the purpose of those designing a class model is different from the purpose of those designing an SQL schema. Whilst those designing a class will focus on the cohesive representation of a network of interacting objects, the focus of those designing a SQL schema is typically data volume, data integrity, and notably the removal of redundant data. A UML class model may only be familiar to one part of a development team: the programmers. Database designers will conceive a different solution based on tables that may not have a one-to-one correspondence to that class model. This difference of focus is the essence of the ownership problem and produces a kind of impedance mismatch that we label an *emphasis mismatch*. An emphasis mismatch is addressed using an ORM mapping strategy.

A mapping strategy is concerned with correspondence between two different descriptions of a concept. In order to address the ownership problem, this correspondence must be documented, published and implemented. Although the detail is application specific, a mapping strategy will generally provide a mechanism for identifying, documenting, and implementing the correspondence of structure and identity between specific entries in a class model and entries in an SQL schema. Hibernate [9] uses XML whilst [22] make use of meta-data stored in SQL tables. This information forms an important part of the design of an object-relational application.

D. Instance

One issue that lies at the heart of ORM practice is the treatment of an object. The problem is that an object is conceptualised as an atomic unit when in practice it has a number of subdivisions. A Java object has subdivisions of structure, state and behaviour. The schema and instance levels of our conceptual framework show how these subdivisions are fragmented (Figure 2). The structure of an object is defined both in a class and an SQL schema (the ownership problem), the behaviour of an object is defined in a class and a valid state of an object must be maintained and enforced both in-memory and across one or more rows in one or more tables, giving rise to encapsulation and identity problems.

In practice fragmentation is addressed using a transformation but there are problems. Data about an object may not transform cleanly to a row of a table or an individual slot ([20],p3) (the instance problem). The structure of an object may not transform to a single table (the structure problem). The SQL-92 standard does not support the behavioural aspects of an object and so the behaviour of an object must be implemented within a Java class at the schema level. The later introduction of persistent stored modules in SQL provided an opportunity for the fragmentation of behaviour. At run-time it may not be necessary to retrieve all the data about an object for a user to complete a transaction. This combined with fragmentation of the universe of objects required to complete a transaction, is the essence of the processing model problem and provides another driver for ORM transformation activities. Fragmentation in the implementation of an object is a significant characteristic of an impedance mismatch. We label this kind of impedance mismatch an *instance mismatch* and it is addressed using an ORM transformation strategy. A programmer must reconcile fragmentation when developing an object-relational application.

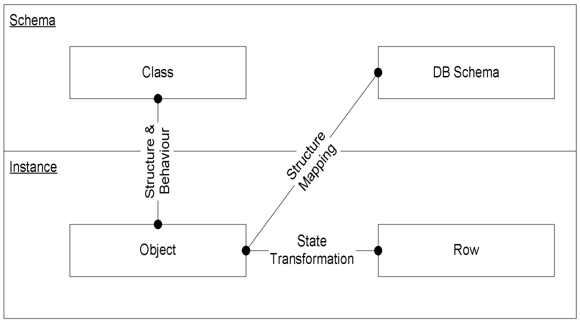


Figure 2. Fragmentation of the subdivisions of an object

The degree of state fragmentation that is characteristic of an instance mismatch is influenced by the ORM mapping strategy employed to produce the structure of an SQL schema. The design of that SQL schema is influenced by the ORM pattern strategy chosen to address a representation mismatch. Choices made within a level of our framework therefore have consequences in other levels.

An instance mismatch transformation strategy must address the consequences of fragmentation in behaviour and state. Such a strategy must deal with the processing model, encapsulation, and instance problems. The SQL standard does not provide support for the definition of behaviour within an SQL schema although relational database vendors have provided such facilities for some years. The valid state of object data may be enforced by rules defined within a class method or as a database constraint. Ambler ([5],p228) describes shadow information that is one strategy for the fragmentation of state, and scaffolding attributes that are one strategy for the fragmentation of structure.

The novel perspective provided by our framework produces new insights in areas such as how to exploit the strategic options available when translating a concept between paradigms and latent issues in solutions that cross over levels of abstraction. In providing an understanding of the issues with an ORM strategy based on levels of abstraction, our framework should provide standards bodies, tool vendors, designers and programmers with new insights into how to address problems of impedance mismatch both at the most appropriate level of abstraction and in the most appropriate way.

VI. A Framework Based Approach

One objective of our framework is to understand the issues and implications of a particular ORM strategy ("strategy"). Our framework does not assume that an object model drives the development of a database schema, nor does it prescribe where to start the analysis of a strategy. In the rest of this paper we show how to use our framework to understand the issues and implications of a strategy and what we can do about them.

Figure 3 is an outline of a four-stage process that provides context and guidance for the use of our framework. Our framework is concerned with the artefacts of object-relational design. The process uses our framework to identify issues with a strategy and to frame solutions to these issues. As such the process augments any software development cycle at the point where a choice of strategy must be made.

The process provides guidance for a change in the way we think about a strategy. Following the process shifts our thinking about a strategy from issues of implementation within the ORIM problem space into the new space provided by our framework. Our framework asks that we think about a strategy in terms of different levels of abstraction. This perspective facilitates new insights into a strategy, an understanding of cause and effect, and suggestions for improvements at the most appropriate level of abstraction.

Our process starts with the strategies in the ORIM problem space. Each strategy addresses one or more problems in the implementation of an object-relational application (Table I). The existence of a problem is the main driver for the use of a strategy. The literature provides some guidance on a choice of strategy based on costs and benefits. Future choices will also be informed by the outcomes from using our framework. The process then proceeds clockwise through the stages of comprehending a strategy, analysis of that strategy, understanding cause and effect in relation to issues with that strategy, and finally reflecting on the issues and suggesting changes to the strategy or the context in which it operates.

In the following sections (A through D) we describe each of the stages of our process. We show that using our process to understand available strategies facilitates a more informed choice of strategy. The objective of the first stage of our process is to comprehend a chosen strategy.

A. *Comprehend a Strategy*

The issue to be explored is how a strategy achieves its objective. In the first instance this comprehension will be based on the published literature and practical experience. We illustrate our approach using a case study that provides a context for strategy analysis. A case study helps clarify the semantics of a strategy, explain issues and highlight outcomes. Applying a strategy to a case study provides a worked example, demonstrates comprehension and cements understanding. A case study and worked example also provide material for illuminating issues in other stages of the process. Once we have an understanding of a strategy our process asks that we now move from the ORIM problem space and think in terms of our framework. In the next stage of the process we use our framework to analyse a strategy.

B. *Analyse a Strategy*

The objective is to provide new insights into a strategy. Issues to be explored include: whether a strategy is consistent in terms of our framework, whether a strategy correctly represent a data structure and its semantics, and whether the assumptions a strategy involves are safe assumptions to make. The resulting issues are then structured in terms of our framework.

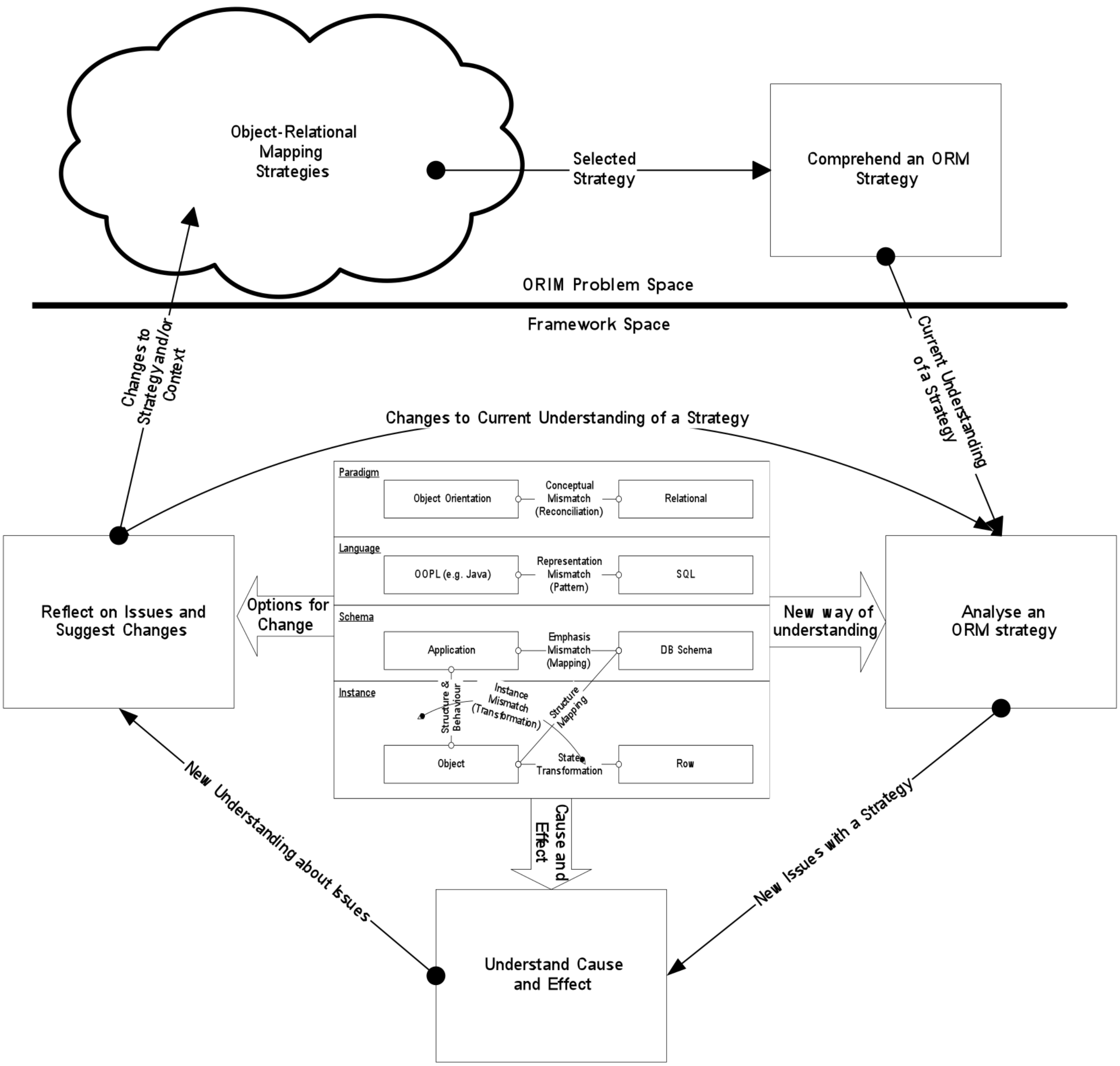


Figure 3. A Framework Based Approach

The issues must be phrased using terms at the appropriate level of abstraction. Each level provides a different focus for analysis and hence, a different set of terms (Table III). For any discourse between silos to be consistent and valid it is important that the corresponding set of terms are used. In the next stage of our process we identify the cause, effect and consequences of these issues.

TABLE III. GUIDELINES FOR THE TERMS USED WITHIN EACH LEVEL OF OUR FRAMEWORK

Level	Terminology
Conceptual	Terms relating to a particular world-view, irrespective of how it is actually described or implemented. Example terms include class, object, message, relation, tuple and union.
Language	Terms relating to language semantics, syntax and grammar, irrespective of a design. Example terms include UML class, Java Class, SQL table and column.
Schema	Terms relating to specific design choices including anything from a universe of discourse. Example terms include Order, Customer, Trade and Equity
Instance	Terms relating to data values. Example terms include instance, row, value and cast.

C. *Understand Cause and Effect*

Our framework is used to provide structure both to the analysis and the results. Issues to be explored include: whether an issue is related to the strategy or the context in which the strategy operates, whether the issue is local to a particular level, and if not what is the cause of an issue. An issue at the schema level may for example, be caused by an omission at the language level. This omission may be a limitation of a particular language or it may be caused by a conceptual difference. Such a conceptual difference would be beyond the scope of an object-relational application project to resolve. In order to correctly address a conceptual difference, the discourse would need to involve at least product vendors, standards bodies and possibly research bodies. Our framework provides the structure necessary to correctly identify and communicate both the cause and the consequences.

D. *Reflect on Issues and Suggest Changes*

Once we understand cause, effect and consequences we are in a position to suggest improvements to a strategy or to the context in which that strategy operates. In the final stage of our process we use the framework to identify options for change. Each level of our framework provides an opportunity to address an issue in a different way. The issues to be explored include: whether it is appropriate to make an improvement at a particular level, what change we need at that level in order to resolve an issue, and whether we change the strategy such that an issue is avoided. In order that others benefit and to avoid wasted effort, suggestions and improvements should be fed back into the wider discourse through changes to ORM tools, standards and the patterns used to describe strategy. Through the use of common abstractions and consistent terminology, our framework provides the structure necessary to communicate these improvements and to facilitate a coherent discourse for their implementation across cultures [23].

In the next section we provide an example of how the process and framework are used together to understand and improve a strategy.

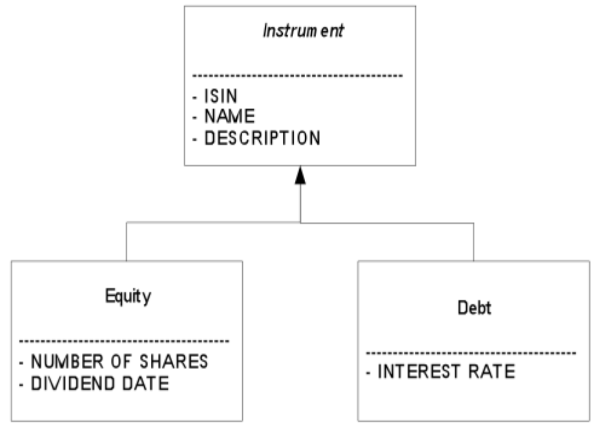


Figure 4. Financial Instrument Class Hierarchy

VII. USING OUR PROCESS – A WORKED EXAMPLE

Figure 4 presents a small class hierarchy for a financial instrument that provides a case study for our worked example.

There are two distinct and mutually exclusive kinds of instrument: Equity and Debt. Each is identified by an International Securities Identifying Number (ISIN) code. The ISIN code is defined under ISO 6166 and is unique across all financial instruments. In order to simplify the example, no associations or aggregates have been used.

We anticipate that such a hierarchy would form the basis for a Java application that would maintain data about financial instruments. The design of that Java application is beyond the scope of this paper but for now we will assume that Figure 4 provides a suitably accurate description of the class model.

Our requirement is to provide a means to store data about the objects of class Equity and class Debt in a relational database. We need to produce a data structure using SQL that corresponds to a Java data structure based on Figure 4. There are a number of strategies that take as their starting point a class hierarchy and produce an SQL-92 based representation. Three such strategies are [13] p9-17:

a) *A single table per class hierarchy*

b) *A single table per concrete class*

c) *A single table per leaf class*

Let us consider strategy (a) ("the strategy"). This strategy combines the definition of all classes in a hierarchy to form a single SQL table. A row of this table will store data about an instance of a class in the hierarchy. We are considering this strategy because Keller [12] recommends it as a strategy for a small application and

Ambler [24] recommends it for systems with a shallow class hierarchy. Ambler [5] suggests that during the development of an object-relational application refactoring is used to implement a change of strategy should it prove necessary. We will use SQL-92 for our example because no description of this strategy uses the additional facilities available in later versions of SQL.

In the following sections we show how in the context of our process, our framework is used to understand and improve the strategy. The outcomes of using our strategy could be used to compare strategies in order to choose the most suitable one. We do not show such a comparison in this paper but focus instead on improving a strategy.

A. Comprehend the Strategy

In the previous section we established our rationale for using the strategy. Here we show how the strategy achieves our objective to store object data.

1) Applying the Strategy to our Case Study

The process of applying the strategy is summarised as follows:

- Create a single table (Ambler suggests using the name of the root class as the table name).
- Create a column for each attribute.
- The data type of a column must correspond to the type of an attribute in so far as it must accept all possible values of that attribute.
- Each column representing a subclass attribute must accept NULL regardless of its definition in the class model.

Applying the strategy to Figure 4 produces the SQL-92 table definition presented in Figure 5. Note that a single row in this table will represent data about either an object of class Equity <u>or</u> an object of class Debt. The columns NUMBER_OF_SHARES, DIVIDEND_DATE and INTEREST_RATE must therefore accept NULL even though for their respective classes they are mandatory.

2) Assumptions

Descriptions of the strategy in Keller [13] and Ambler [24] make the following assumptions:

- It is *not* necessary to maintain the parent-child relationship between a class and a subclass in the database. This relationship is used to identify the attributes necessary for the definition of a table.
- An object can be fully described using a single row.
- The data types of a class attribute and a column are compatible.
- Only that column corresponding to an attribute of a class to which an instance belongs is set for a row. All other columns will be set to NULL.
- The mapping of a class attribute to a column is documented in some form or at least is somehow known by those who must use it.
- If the data type of a class member attribute is changed, regardless of the topological position of the class in the hierarchy, that change applies to all rows of the table.

3) Costs and Benefits

The main benefits of this strategy are [24]:

- Data about all objects is accessible from a single table;
- There is only one table for a programmer to consider;
- The mapping from a class hierarchy to a single table is a "simple approach";
- It is easy to add a new class should requirements change.

```
create table INSTRUMENT(
   ISIN CHARACTER(12) PRIMARY KEY,
   NAME CHARACTER(20),
   DESCRIPTION CHARACTER(40),
   NUMBER_OF_SHARES INTEGER NULL,
   DIVIDEND_DATE DATE NULL,
   INTEREST_RATE FLOAT NULL)
```

Figure 5. The SQL-92 table derived from the Instrument class hierarchy

Ambler and Keller describe a number of issues with this strategy. One such issue relates to classification. In order to maintain the class member semantics of a collection of data values in the context of table INSTRUMENT, there must be some means to differentiate between data for an object of class Equity and data for an object of class Debt. There are at least three options for achieving this:

a) Infer the class of a row from the existence of values for its attributes [13], p13. For example only the row for an object of class Equity will have a value for NUMBER_OF_SHARES. For a Debt object this column would have a NULL value.

b) Augment the table definition with a new column the value of which indicates the class of data to which a row belongs [24]. For a row representing data about an object of class Equity for example, this column would have the value "EQUITY".

c) Use a discriminator value from the universe of discourse in order to differentiate the class of data stored in a row [13], p13. Similar to option "a" but here we look at the actual value not its presence, and option "b" but use the values stored in an existing column rather than creating a new one. All Debt ISIN codes could include the character "D" at a certain position. This character indicates that the ISIN code is non-atomic and identifies data about an object of class Debt.

Let us consider option (a) because it does not require the maintenance of additional data. A user of the table INSTRUMENT must know how to infer class membership.

Other issues documented in [13] and [24] include potential wasted space in the database through the use of NULL, the consequences of certain changes to the class hierarchy, locking issues because all data is the same space, and indexing issues because secondary indexes are required.

With as full grasp of the current knowledge about a strategy as space allows, and a worked example based on a case study to cement this understanding, we are now in a position to start our analysis. This analysis will identify the cause of these implementation issues and highlight new issues.

B. *Analysis of the Strategy*

The observations and insights we have gained during our analysis of the strategy are described in the following subsections. They are not listed in any particular order but they are categorised using the levels of our framework.

1) *Conceptual Insights*

We need to represent a class hierarchy in a SQL schema in order to provide for requirements and design traceability. It is essential to understand the semantics of a particular class hierarchy before applying the strategy. The strategy does not make clear which definition of a class is being used.

In mapping a class to a table, the strategy mixes levels of abstraction (the shaded boxes in Figure 6). The term "table" is a language level construct within the relational silo. The term "class" is a conceptual level construct within the object silo. This strategy should either map a class to a relation or map a class in a particular object-oriented programming language, e.g., Java to a table. This is an important distinction because the semantics of a Java class are not the same as those of a C++ class at the language level. A C++ class for example supports multiple inheritance.

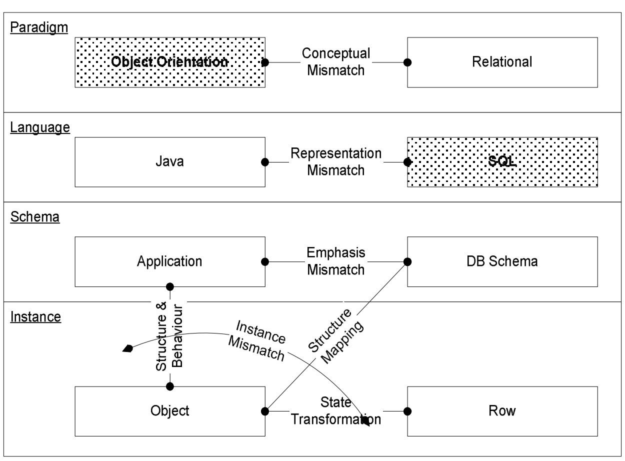


Figure 6. Mixing levels of abstraction

A relation represents a kind of fact. In combining the definitions of all subclasses into a single relation this strategy overloads the semantics of a relation. A relation must now represent more than one kind of fact although each tuple represents a single fact.

2) *Language Insights*

The strategy employs a class hierarchy as the basis for the definition of a table, but the actual hierarchy is not represented in the database. We therefore lose requirement and design traceability.

SQL-92 provides no explicit support for a hierarchy in the definition of a table. Support for a hierarchy can be designed into a table but the strategy does not require, nor the SQL-92 based representation (Figure 5) preserve, the parent-child relationship between a class and a subclass.

To ensure data integrity and to enforce the semantics of a disjoint subclass, there must be some way of identifying to which class the data in a row belongs.

A column in a table represents an attribute of a class. The assumption is that they are of equivalent data types. SQL-92 has a predefined set of data types. The type of a class attribute may be another class although that is not shown in this example. The definition of that class is a schema level decision so there is no guarantee of type compatibility at the language level and casting must be used. The strategy does not describe how to address differences in type or scale.

3) *Schema Insights*

The classes Equity and Debt are disjoint. Class Instrument is abstract. These are design features built into our class model (Figure 4). The SQL table INSTRUMENT is formed from the union of the attributes of the class Instrument and the subclasses Equity and Debt. These attributes are represented as columns of the table INSTRUMENT. Data about each object is stored in a row of the table INSTRUMENT. As the primary key, ISIN provides the semantics of a disjoint subclass because it is unique across all financial instruments.

Instrument is an abstract class. Although no object of this class *should* exist, for reasons of data integrity it is important to prevent the insertion of a row of this class in the table INSTRUMENT.

The data type of each column of the table INSTRUMENT has a trivial correspondence to the type of the corresponding attribute of each class. This correspondence is not always so trivial. Some attributes may be derived or use names which are not the same as or similar to the column name. A user of the schema must understand that a row of the table INSTRUMENT represents data about one of two kinds of object. They must also know how to differentiate those kinds of data.

4) *Instance Insights*

The class of data stored in a row may be determined in a number of ways. The choice must be made clear to those who use the table. In this example we have chosen to infer the class from one or more column values.

The semantics of the Instrument class hierarchy are not represented in the table INSTRUMENT and so it is not straightforward to query over a subclass of Instrument and all its subclasses. In our simple example classes Equity and Debt do not have a subclass. If they did a programmer must understand the conditions for returning only those rows belonging to each object of each subclass in which they have an interest.

TABLE IV. INSIGHTS INTO THE STRATEGY FROM USING OUR FRAMEWORK

Level	Insights into the Strategy
Conceptual	Mix levels of abstraction. A relation has no explicit semantics of hierarchy. Overload the semantics of a relation.
Language	Omit the subclass relationship. Issues of type or scale between an attribute and a column.
Schema	Enforce the semantics of a subclass. Enforce the semantics of an abstract class. Make explicit the correspondence between attributes and columns. Differentiate the class of data held in a row.
Instance	Identifying the class to which the data in a row belongs. Query a sub-hierarchy.

5) *Summary*

We have used our framework to question the strategy at a number of levels of abstraction. Table IV summarises the insights that thinking about the strategy in terms of our framework provides. Each level of our framework has focused attention on a different aspect of the nature of the strategy. Our framework has helped us to see new issues and relationships. In the next section we use our framework to explore these relationships and their consequences.

C. *Understand Cause and Effect*

Here we provide two examples of cause and effect relationships based on the conceptual issues of overloaded semantics and support for the semantics of hierarchy.

1) *Overloaded Semantics*

In Table V we use the levels of our framework to show the consequences of overloading the semantics of a relation.

The strategy is described in [13] p9-17 using terms that we generally recognise within the language level of our framework, for example a class corresponds to a table and an attribute corresponds to a column. This strategy does not address the root cause of this problem only the symptoms. Our framework shows that we must look to the conceptual level for the cause of the overloaded semantics problem.

The results of applying our framework (Table V) clearly show that the conceptual problem of representing more than one kind of fact using a relation has consequences within the levels below. The strategy confronts this conceptual problem at the language level by requiring a way to differentiate the data stored in a row. The choice of mechanism for differentiation will impact the definition of the table INSTRUMENT at the schema level. Ultimately at the instance level, a programmer working with data in a row of table INSTRUMENT must understand how to differentiate between data about an Equity object and data about a Debt object.

Using our framework we relate the consequences of this conceptual problem back to the implementation problem of wasted space described by Amber and Keller (see Section VII.A). The overloading of a relation necessitates NULL valued columns. WHERE clause complexity is another consequence of overloading the semantics of a relation.

TABLE V. OVERLOADING THE SEMANTICS OF A RELATION

Level	Consequences
Conceptual	In our example a relation must represent two kinds of fact. There must be some way to differentiate the class of data held in a relation.
Language	An SQL table is a representation of a relation. The SQL language requires that all rows in a table share the same definition provided by that table. The strategy necessitates we compromise by (i) providing some way to differentiate a row, and (ii) accepting that a column corresponding to a subclass attribute must accept NULL. Preservation of semantics requires that the SQL language support a form of constraint. There must be a mechanism to document the correspondence between a column and a class attribute.
Schema	The definition of table INSTRUMENT must provide some way to differentiate a row. A row may represent data about an object of class Equity and an object of class Debt. We chose to infer class membership from column values. The columns NUMBER_OF_SHARES, DIVIDEND_DATE and INTEREST_RATE must accept NULL even though for their respective classes they are mandatory. A database constraint must ensure attributes are populated correctly based on class.
Instance	Our chosen option for identifying class membership does not require additional columns but class membership is not explicit. A user of the table INSTRUMENT must understand how to differentiate the class of a row based on the value or one or more columns. Differentiating a row based on multiple columns adds complexity to a WHERE clause. Using another method for identifying class membership requires more data be maintained but would make class membership more explicit. A program must ensure that the correct columns are populated for each subclass. A DBA must enforce this using a constraint. The use of NULL values can result in wasted space in a database but this depends on vendor implementation.

2) *Omitting the Semantics of Hierarchy*

In Table VI we use the levels of our framework to show the consequences of omitting the semantics of hierarchy.

We must look to the conceptual level of our framework for the root cause of the hierarchy problem. The semantics of hierarchy are not present in a relation. The strategy does not attempt to address this problem at the language level. As a result at the schema and instance levels it is necessary to encode the semantics of a hierarchy outside the table INSTRUMENT.

A consequence of omitting the semantics of hierarchy is that these semantics are encoded in database constraints and in each query that needs to make use of them. Should the hierarchy change, all places where these semantics are encoded must also be identified and changed. We must encode the semantics of hierarchy because they are not present in the table INSTRUMENT. They are not present because an SQL table does not have explicit support for a hierarchy and the strategy does not address this. An SQL table has no support for hierarchy because it is based on the concept of a relation which itself supports no notion hierarchy. If we had adopted the approach of [3] and focused solely on aspects covered by the schema an instance levels of our framework, we would not have identified the real cause of this problem.

We have identified new issues, traced their cause, and shown that they have consequences for those developing an object-relational application. Our framework can also be used to understand an existing issue and to provide one possible chain of cause and effect. Table VI shows that whilst providing a means to query the Instrument hierarchy, this strategy introduces problems if one wishes to query any hierarchy below that (an issue not present in our case study).

TABLE VI. OMITTING THE SEMANTICS OF HIERARCHY

Level	Consequences
Conceptual	The semantics of a class hierarchy are well defined, but the actual semantics in use depend on the context provided by a class model and the intention of the designer. A relation has no explicit semantics of hierarchy.
Language	An SQL-92 table also has no explicit semantics of hierarchy. The strategy does not provide a means to address this.
Schema	The semantics of the Instrument class hierarchy are not present in the table INSTRUMENT. In order to preserve data integrity, a DBA must encode these semantics in one or more database constraints.
Instance	In terms of the class hierarchy, all we can say about a single row is that it belongs to a given class and to the hierarchy rooted at class Instrument. Information regarding the topological position of that class in the hierarchy is not present in either the data or the definition of the table INSTRUMENT. A position may be inferred [25] but this should not be necessary and is prone to ambiguity. As a consequence, to correctly form a polymorphic query over a sub-tree, a programmer must encode the semantics of the Instrument hierarchy in a query. The deeper the hierarchy one represents using a single table, the more complex the WHERE clause becomes. This is particularly true if one wishes to query data for objects belonging to a leaf class.

Our framework can also be used to clarify received wisdom. Contrary to Keller's suggestion in [13], p13, it is not sufficient to only identify to which class a row of data belongs. The query must also include the semantics of that hierarchy. Table VI uses our framework to show why a query must include these semantics.

The root cause of an issue is not always at the conceptual level of our framework. We have assumed that there is a direct correspondence between a class hierarchy and a table. A schema provides the context necessary for normalisation. Normalisation is a process within the relational silo that breaks down correspondence at the schema level. This issue must be resolved within the design of a schema.

3) Summary

We have shown that our framework provides a way to understand both the cause of an issue with a strategy and the consequences of that issue. The root cause of an issue may be at any one of the levels of our framework and its effect may materialise in different ways. In the next section we use our framework to reflect on this new understanding and suggest opportunities for improvement.

D. Reflect on Issues and Suggest Changes

We have identified two issues with the strategy: overloading the semantics of a relation, and omitting the semantics of hierarchy. We can improve the strategy in two ways: either indirectly by addressing the symptoms of an issue or directly by addressing the context.

The context of any given level of our framework is those levels above it, so for the schema level the language and conceptual levels provide the context. The cause of an issue may be at any level of our framework. Our framework also provides a means to understand at which level symptoms emerge and for thinking about the most appropriate approach to address them.

Understanding cause and effect is not the only requirement for change. The ability to effect change depends on the power and influence of those involved. Ideally the root cause of an issue should be addressed, but this is not always an option for those developing an object-relational application. Their influence will typically concern the schema and instance levels although the use of dynamic languages such as Ruby and Groovy [26] may change this. If an issue is best resolved at the conceptual or language levels they will still have to adopt an indirect approach and therefore only address the symptoms of an issue. Those involved with the definition of a standard or the design of a programming language will have influence to affect change at the language level. Research bodies and the community in general are best placed to deal with a conceptual issue. They have the power and influence to adopt a direct approach.

1) Indirectly

An indirect approach takes context as given and will not address the root cause of an issue. A solution at the schema level must work within the constraints of the languages used and as a result also accept any conceptual problems. The root cause of both our issues is at the conceptual level. A direct approach in this case will therefore involve avoidance or mitigation at best. Table VII summarises some of the indirect options available for addressing aspects of each issue.

Ultimately it may be more appropriate to use a different strategy. In order to address the first issue we could use a strategy that involves creating a separate relation for each concrete class [24]. This would remove some of the WHERE clause complexity in terms of class identification and joins, wasted space and the need to maintain additional data but would go against the spirit of the strategy: to represent all data in a single table.

TABLE VII. INDIRECT OPTIONS

Level	Suggestions
Schema	Use a different strategy. One that produces a separate relation for each concrete class. Create a database view for each class. Add a column PARENT_CLASS that indicates the parent class.
Instance	Using a different strategy avoids maintenance of additional data. Use a database view to realise data for a subclass. Infer class membership from attribute values. Set the value of PARENT_CLASS to be the classifier value for the parent class.

One solution that does not involve a change of strategy is to retain the single base table INSTRUMENT but represent each subclass or subclass hierarchy as a database view. Using a database view *hides* WHERE clause complexity for a schema user and the semantics need only be defined in one place. This solution does not address the need to maintain additional data or the problem of wasted space in the base table (although this is arguably a database vendor implementation issue). The use of a database view would increase the space required but only marginally if a materialised view is not stored. We can avoid the maintenance of additional data if we continue to infer class membership from the existence of data values or use an existing discriminator value from the universe of discourse.

Neither approach addresses the omission of the semantics of hierarchy. Adding a column PARENT_CLASS does not solve the problem because it confuses intent and extent. The semantics of a hierarchy are mixed with the data representing an object. This messy implementation fudge is not a viable solution because it is still necessary to know how the hierarchy is structured and there are problems with an abstract class or any class where no rows (yet) exist.

2) Directly

Here we use the levels of our framework to suggest changes to the context in which the strategy operates. In Table VIII and Table IX are options for addressing both issues at each level of our framework. We do not propose a complete solution. Our objective is two fold. First to show that there are options at the conceptual and language levels, and second to highlight that these provide different options at other levels of our framework.

In the case of both issues, the root cause of the problem is at the conceptual level of our framework. This is therefore the most appropriate level at which to make improvements, but changes at this level are the most fundamental. A change at the conceptual level will have far reaching consequences, will require input from researchers and standards bodies, and consequently will take time to implement. Such a change is out of scope for any object-relational application development project.

We note that work to address these issues by changing context has already started. The majority of the solution described in Table IX is possible using the object-relational features introduced in SQL:1999 (“OR-SQL”). Only the ability to insert data into the table INSTRUMENT and have a row created in an appropriate sub-table is not supported. Although counterintuitive, this facility may be important for a programmer because it maintains the single table nature of the solution provided by the original strategy.

TABLE VIII. DIRECT OPTIONS FOR ADDRESSING OVERLOADED SEMANTICS

Level	Suggestions
Conceptual	Recognise that a relation may represent more than one kind of fact.
Language	Provide a classifier mechanism in the definition of a table. Extend the SQL language or its implementation to support optional columns based on this classifier.
Schema	Do not represent Equity and Debt as subclasses. Use a single class Instrument. This is not in the spirit of the object model and may cause issues in the object silo. Represent each class using a separate table. Again, this is not in the spirit of the strategy.
Instance	Provide access to the classifier mechanism above within a query. Insert only the data values required based on the classifier. Omit a column if it is not relevant to a particular kind of row.

In Section VII.A.3 we listed some of the benefits of the strategy. These benefits come at a cost. Storing data about all objects in a single table may be a “simple approach” [24] but it has costs in terms of work on database constraints and queries. Whilst it may be easy to add a new class, such a change has consequences including the maintenance of database constraints and queries. Our framework has drawn attention to these problems and provided a way to think about improving the situation. The information emergent from the use of our framework and process is therefore of benefit to those who must choose and implement this strategy.

TABLE IX. DIRECT OPTIONS FOR ADDRESSING THE OMISSION OF HIERARCHY

Level	Suggestions
Conceptual	Recognise the possibility of a hierarchy of relations. Support the concept of an abstract relation.
Language	Support a hierarchy of tables and permit a single query over the hierarchy of tables. That query does not need to include the names of all sub-tables.
Schema	Create a separate table for classes Instrument, Debt and Equity but each table is part of a hierarchy of tables. Each table may be queried individually or as part of a hierarchy.
Instance	Create a row in the corresponding base table or by inserting into table Instrument. Query the entire hierarchy or part thereof using a single statement.

3) Summary

Our analysis has demonstrated that this ORM strategy does not address two conceptual problems because it is a solution at the language level of our framework. The strategy does not attempt to mask these problems and this results in work for those who use it to implement an object-relational application.

There are indirect options open to those developing an object-relational application. Whilst these do not address the fundamental problem they will improve the situation in the short term leaving time for conceptual issues to be addressed through a direct approach.

There are a number of strategies for any impedance mismatch problem. It may be that using another strategy is more appropriate for those developing an object-relational application. We anticipate that effecting change at the conceptual and language levels of our framework will be more difficult than at the schema or instance levels. Changing the definition or implementation of SQL for example is not feasible for those developing an object-relational application. Our framework provides a basis for making the decision to change by asking that we think about cause, effect and consequences. That information helps when selecting amongst alternatives. At this point we have come full circle in our process (Figure 3).

VIII. Conclusion and Future Work

Our conclusions concern the framework used to understand a strategy and the process by which we used the framework to suggest improvements to a strategy.

1) *The Framework*

We have demonstrated that understanding a strategy at different levels of abstraction does identify the root cause of an issue. Our framework is not concerned with the issues of implementation that have driven work by Ambler [24] and Fussell [11]. We have also demonstrated that in order to address an ORIM problem at the most appropriate level of abstraction we must understand the real issues that underpin that problem.

In our framework we have a new way to understand an ORM strategy. If we think about a strategy at a number of levels of abstraction we find new insights into a strategy. These insights provide an opportunity to improve a strategy and the context in which a strategy operates. If the outcomes appear obvious it is because of the new perspective provided by our framework. A perspective that takes context as given, is driven by a single problem, or which views a solution as an exercise in software architecture ([16], [11]) will not produce the same results.

Ambler [16] suggests software architecture as a means to shield a programmer from the details of a strategy. In terms of our framework this is predominantly a schema level activity within the object silo. Fussell [11] suggests a separation based on client and server. This separation corresponds loosely to the object and relational silos of our framework. Fussell's emphasis is on decoupling but impedance mismatch problems occur when we try to combine object and relational artefacts. Neither perspective provides the same scope or a means to facilitate an analysis of cause and effect and an understanding of consequences that we have achieved from the use of our framework. Taking a step back from the detail of implementation, our framework allows us to address the cause of a problem, not its symptoms, at the most appropriate level of abstraction.

The information elucidated through the use of our framework will be of use to standards bodies, tools vendors and those who define a strategy. Thinking about the consequences of a strategy provides information necessary to choose between alternatives. Those working on an object-relational application can now make a more informed choice of strategy. Those working on database and programming language standards see the impact of past choices and the need for change. Researchers in object and relational concepts see the consequences of their work and that there is still work to be done to cross the chasm [27].

The framework helps bridge the cultural impedance mismatch [23]. Through the use of common levels of abstraction our framework facilitates a discourse between proponents of object and relational perspectives. A specific set of terms must be employed at each level of the framework although further work is required to develop a formal ontology of terms based on Table III. We are now in a position to address problems of an ORIM in a structured and consistent way, not just across levels of abstraction but also between silos. We can now think in an integrated way, for example how decisions made in the design of Java correspond to structures in SQL or vice versa. We also have a way to understand the impact of these changes for those designing both an object and a relational schema and programming an object-relational application.

Another opportunity for our framework is to understand the impact and potential of changes introduced in OR-SQL on the current ORM strategies. In terms of our framework, OR-SQL appears to characterise a language level change in the relational silo. Further work is required to understand the opportunities these changes present for new or enhanced ORM strategies with languages such as Java, LINQ [15] and Ruby [26].

A generalised form of our framework could help to understand issues at the junction of any two paradigms in computing or other disciplines.

2) *The Process*

We have demonstrated that our process provides the necessary guidance to improve a strategy. We have identified options for change that are linked to a conceptual problem not a symptom of an implementation. We have also demonstrated that our process supports a shift in thinking away from implementation issues because we start by understanding a strategy and issues of implementation, but finish by suggesting solutions at a number of levels of abstraction.

References

[1] Ireland, C., Bowers, D., Newton, M., Waugh, K.: A Classification of Object-Relational Impedance Mismatch. In: Chen, Q., Cuzzocrea, A., Hara, T., Hunt, E., Popescu, M. (eds.): The First International Conference on Advances in Databases, Knowledge and Data Applications, Vol. 1. IEEE Computer Society, Cancun, Mexico (2009) p36-43

[2] Neward, T.: The Vietnam of Computer Science (http://blogs.tedneward.com/2006/06/26/The+Vietnam+Of+Computer+Science.aspx) (6th February 2007)

[3] Stathopoulou, E., Vassiliadis, P.: Design Patterns for Relational Databases. Vol. 2009. ODMG (2009)

[4] Meijer, E.: There is No Impedance Mismatch (Language Integrated Query in Visual Basic 9). OOPSLA. ACM, Portland, Oregon (2006)

[5] Ambler, S.W.: Agile Database Techniques - Effective Strategies for the Agile Software Developer. Wiley (2003)

[6] Keller, A.M., Jensen, R., Agarwal, S.: Persistence Software: Bridging Object-Oriented Programming and Relational Databases. In: Buneman, P., Jajodia, S. (eds.): ACM SIGMOD international conference on management of data, Vol. 22. ACM Press, Washington, D.C (1993) 523-528

[7] Marguerie, F.: Choosing an object-relational mapping tool (http://weblogs.asp.net/fmarguerie/archive/2005/02/21/377443.aspx) (14th November, 2007)

[8] Holder, S., Buchan, J., MacDonell, S.G.: Towards a Metrics Suite for Object-Relational Mappings. COMMUNICATIONS IN COMPUTER AND INFORMATION SCIENCE **8** (2008) 43-54

[9] Hibernate: (www.hibernate.org)

[10] Biswas, R., Ort, E.: The Java Persistence API - A Simpler Programming Model for Entity Persistence (http://java.sun.com/developer/technicalArticles/J2EE/jpa/index.html) (25th September 2007)

[11] Fussell, M.L.: Foundations of Object Relational Mapping (http://www.chimu.com/publications/objectRelational/) (25th September 2007)

[12] Hohenstein, U.: Bridging the Gap between C++ and Relational Databases. In: Cointe, P. (ed.): European Conference on Object-Oriented Programming, Vol. Lecture Noted on Computer Science 1098. Springer-Verlag, Berlin (1996) 398-420

[13] Keller, W.: Mapping Objects to Tables: A Pattern Language. In: Bushman, F., Riehle, D. (eds.): European Conference on Pattern Languages of Programming Conference (EuroPLoP), Irsee, Germany (1997)

[14] Lammel, R., Meijer, E.: Mappings Make Data Processing Go 'Round: An Inter-paradigmatic Mapping Tutorial. Lecture Notes in Computer Science **4143** (2006) 169-218

[15] Schwartz, J., Desmond, M.: Looking to LINQ (http://reddevnews.com/features/print.aspx?editorialsid=707) (23rd October 2007)

[16] Ambler, S.: The Design of a Robust Persistence Layer for Relational Databases (http://www.ambysoft.com/downloads/persistenceLayer.pdf) (10th May 2007)

[17] Griethuysen, J.J.v. (ed.): Concepts and Terminology for the Conceptual Schema and the Information Base. ISO, New York (1982)

[18] Coad, P., Yourdon, E.: Object Oriented Analysis. Yourdon Press (1990)

[19] Codd, E.F.: A relational model of data for large shared data banks. Communications of the ACM **13** (1970) 377-387

[20] Kalman, D.: Moving forward with relational: looking for objects in the relational model, Chris Date finds they were there all the time. DBMS, Vol. 7 (1994) 62(66)

[21] Meijer, E., Schulte, W.: Unifying Tables, Objects, and Documents (http://research.microsoft.com/~emeijer/Papers/XS.pdf) (21st August 2007)

[22] Sutherland, J., Pope, M., Rugg, K.: The Hybrid Object-Relational Architecture (HORA): an integration of object-oriented and relational technology. ACM/SIGAPP symposium on Applied computing: states of the art and practice. ACM Press, Indianapolis, Indiana, United States (1993)

[23] Ambler, S.: The Cultural Impedance Mismatch Between Data Professionals and Application Developers (http://www.agiledata.org/essays/culturalImpedanceMismatch.html) (10th May 2007)

[24] Ambler, S.: Mapping Objects to Relational Databases: O/R Mapping In Detail (http://www.agiledata.org/essays/mappingObjects.html) (12th April 2007)

[25] An, Y., Borgida, A., Mylopoulos, J.: Discovering the Semantics of Relational Tables Through Mappings. LNCS 4244 - Journal on Data Semantics VII (2006) 1-32

[26] Richardson, C.: ORM in Dynamic Languages. Communications of the ACM **52** (2009) 48-55

[27] Brown, K., Whitenack, B.G.: Crossing Chasms: A Pattern Language for Object-RDBMS Integration "The Static Patterns" (http://www.ksc.com/articles/staticpatterns.htm) (30 December 2008)

Efficient XML data management for systems biology: Problems, tools and future vision

Lena Strömbäck, David Hall, Mikael Åsberg
Department of Computer and Information Science
Linköpings Universitet
Linköping, Sweden
Email: lestr, g-davha, g-mikas@ida.liu.se

Stefan Schmidt
Institute of Computer Science
Rostock University
Rostock, Germany
e-mail: mail.stefan.schmidt@googlemail

***Abstract*—Recently, XML has become a very popular representation format for exchange of data within systems biology. This has made large amounts of XML data available on the Internet and there is a need for tools to easily and efficiently manage this data. In this paper we give an overview of existing standards and analyze the situation. We describe two tools that have been developed to provide and experiment with data management for XML standardized data. We evaluate the efficiency for each of the tools, show that they provide more efficient data management and make a proposal for a future combined solution. The paper is an extended version of [1] where we put the work in a larger context of efficient XML data management for systems biology.**

***Keywords*-XML; XQuery; hybrid XML management; graph processing; systems biology**

I. Introduction

During the past few years researchers within bioinformatics and systems biology have started to produce larger and larger quantities of experimental data. The goal in the area is to understand how proteins, genes, and other substances interact with each other within living cells. This is the key to understand the secret of life, and as such it has been set as a major goal for bioinformatics research by the Human Proteome Organization [2] and the US National Human Genome Research Institute [3]. Enhanced understanding in this area is essential for discovering new medical treatments for many diseases.

Within the area the tradition has been to publish results from experiments in databases on the web [4], [5], [6], [7], [8], making it possible for researchers to compare and reuse results from other research groups. The information content, data model and functionality are different between the databases, which makes it hard for a researcher to track the specific information he or she needs. However, most of the databases provide some kind of export facility in one or several XML-based exchange formats for protein interactions, e.g. SBML [9], PSI MI [2], and BioPAX [10].

One important discipline within systems biology where many standards exist and the emphasis of this article are biological pathways and molecular interactions. In this area the data form complex networks and it is important to enable analysis of these networks to detect key molecules for functionality or similarities between different species [11], [12].

One reason for the popularity of XML for exchange of data within bioinformatics and other areas is its flexibility. XML can be used for representing all kinds of data ranging from marked-up text, through so called semi-structured data to well structured datasets. This is a benefit especially within systems biology where datasets often contain well structured parts, such as tables or interaction graphs and unstructured or semi structured annotations or descriptions of, for instance, the experimental setup.

Supporting the flexibility that makes XML appealing is challenging from data management and technical perspectives. Two main approaches have been used, native databases designed specifically for XML and shredding XML documents to relations. More recently, hybrid implementations that combine native and shredding strategies are provided by the major relational database vendors (Oracle www.oracle.com, IBM www.ibm.com/db2 and Microsoft www.microsoft.com/sql/default.mspx). This offers new options for storage design where native and relational storage can be used side by side for different parts of the XML data. Within systems biology the situation is further complicated by the need for graph analysis functionality, which requires complex analysis capacity.

In this paper we will further analyze the situation and present two tools for management of XML data within bioinformatics. The paper starts with a brief overview of availability of standards and data within bioinformatics. Based on this overview we present the goals and motivations for the work. We then present two different tools. The first is a graph analysis extension to XQuery that enables efficient and easy to use graph functionalities. The second is a tool that enables the user to design and compare hybrid XML storage and thus further improve efficiency of the storage model. For each tool we present the main ideas behind them, exemplify the use of the tool and evaluate the performance. At the end of the paper we discuss related work and lay out the direction of a full scale future tool that could be

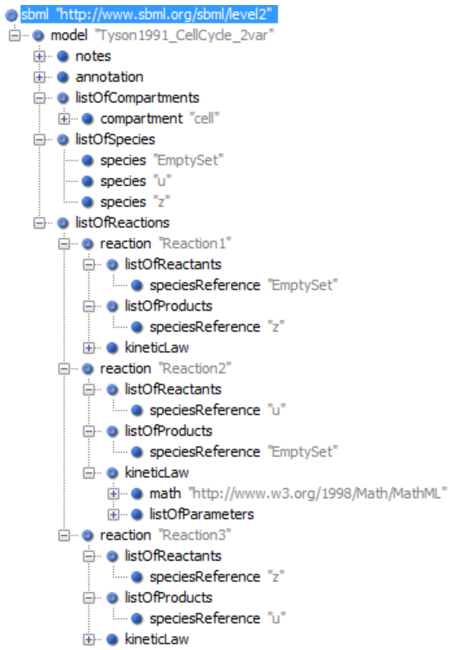


Figure 1. Excerpt of an SBML document.

supportive of data management within bioinformatics.

II. XML STANDARDS FOR BIOINFORMATICS

In a study [13] in 2006 we found 85 XML-based formats in systems biology. These include formats for exchanging information about substances, interactions, pathways, compartments, organisms and experiments.

With the large interest in using XML-based formats for exchange and export of data within systems biology the need for standardization has become obvious. Some formats have become de facto standards or at least widely accepted formats (for example Seq-entry and INSD-Seq [14]), while other are intended as candidates for future standards. Table I is based on the evaluation in [13] and lists examples of commonly used XML-based bioinformatics formats. The version given for each format is the latest version available. However, in many cases, actual use and support in software and databases may be predominant for earlier versions.

Of the formats listed there are formats for representing molecular interactions or pathways, describing structure of substances (DNA, RNA, proteins or other chemical compounds). The formats for interactions and pathways could be either aimed at describing simulation properties (e.g. SBML[9] or CellML[15]) or experimental results (e.g. PSI MI [2]). The formats for structure of substances are often export formats for certain databases.

Figure 1 shows the basic structure of an SBML document. It contains lists of compartments, species and reactions that are part of the simulation model. Internal references are used to connect species to reactions, thereby avoiding redundancy

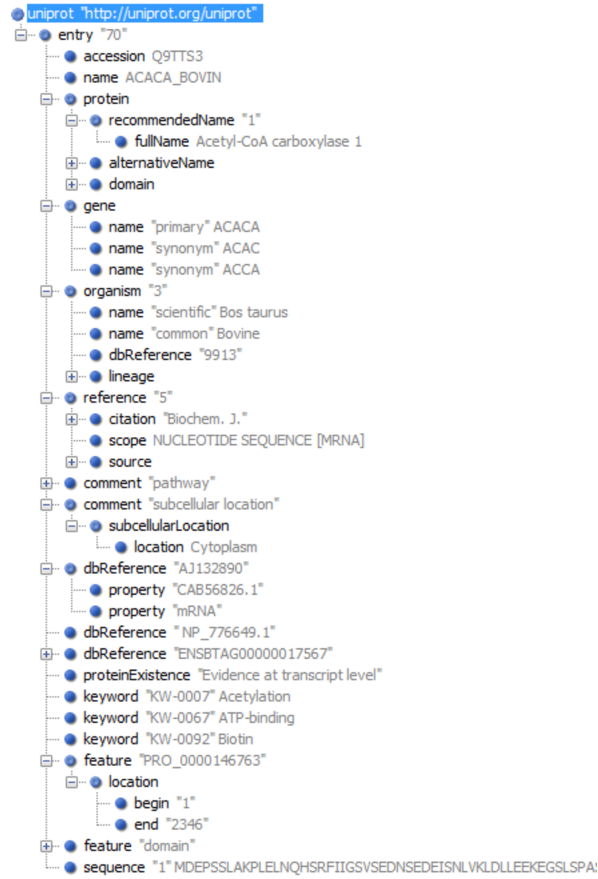


Figure 2. Excerpt of an UniProtKB document.

of species information. Figure 2 shows the basic structure of a UniProtKB document. It contains a list of entries which in turn contains elements with name information for proteins, genes, and organisms, database and literature references, and additional information (annotations). The entries also contain (not depicted in the figure) sequence (for the protein) and keywords (using controlled vocabularies). Here emphasis is on citations, names and taxonomy.

During the latest years efforts to standardize XML-based formats in the bioinformatics area has been intensified. Organizations such as the Proteomics Standards Initiative (PSI) and Institute for Systems Biology (ISB) have developed standards within different fields of bioinformatics. Adoption of standard formats is delayed due to implementation in tools and database APIs/data dumps.

Sometimes several standard formats for the same type of information are developed. In the mass spectrometry area standardization attempts led to mzData[20] (PSI) and mzXML[19] (ISB), both of which are supported in different tools. The two organizations has been working on a joint standard, mzML[23], that combines aspects of mzData and mzXML and version 1.0.0 was released in June 2008 [24]. Another release, 1.1.0, was made in 2009 [24] for fixing

Name	Ver.	Year	Defined by	Purpose	Data
SBML [9]	2.4	2008	Systems Biology Workbench development group.	A computer-readable format for representing models of biochemical reaction networks.	Data available from many databases, for instance, KEGG, www.genome.jp/kegg/ and Reactome, www.reactome.org.
PSI MI [2]	2.53	2006	HUPO Proteomics Standards Initiative.	A standard for data representation for protein-protein interaction to facilitate data comparison, exchange and verification.	Datasets available from many sources, for instace IntAct www.ebi.ac.uk/intact/, and DIP http://dip.doe-mbi.ucla.edu/.
Bio-PAX [10]	L. 3 (0.92)	2008	The BioPAX group.	A collaborative effort to create a data exchange format for biological pathway data.	Datasets available from Reactome www.reactome.org
CellML [15]	1.1	2002	University of Auckland and Physiome Sciences, Inc.	Support the definition of models of cellular and subcellular processes.	CellML Model Repository (240 models) www.cellml.org.
CML [16]	2.2	2003	Peter Murray-Rust, Henry S. Rzepa.	Interchange of chemical information over the Internet and other networks.	BioCYC www.biocyc.org.
EMBL-xml [14]	1.1	2007	European Bio-informatics Institute.	More stability and fine-grained modelling of nucleotide sequence information.	EMBL www.ebi.ac.uk/embl .
UniProt KB [17]	1.28	2009	UniProt Consortium	XML Schema for UniProtKB	Swiss-Prot and TrEMBL www.uniprot.org
INSD-seq [14]	1.5	2009	International Nucleotide Sequence Database Collaboration	The purpose of INSDSeq is to provide a near-uniform representation for sequence records.	EMBL www.ebi.ac.uk/embl and GenBank www.ncbi.nlm.nih.gov/Genbank .
Seqentry	n/a	n/a	National Center for Bio-technology Information.	NCBI uses ASN.1 for the storage and retrieval of data such as nucleotide and protein sequences. Data encoded in ASN.1 can be transferred to XML.	Entrez www.ncbi.nlm.nih.gov/Entrez.
MAGE-ML [18]	1.1	2003	Microarray Gene Expression Data.	To facilitate the exchange of microarray information between different data systems.	ArrayExpress www.ebi.ac.uk/arrayexpress.
Mz XML [19]	2.1	2004	Institute for Systems Biology	The common file format for mass spectrometry data.	PeptideAtlas www.peptideatlas.org, Sashimi sashimi.sourceforge.net, Open Proteomics Database http://apropos.icmb.utexas.edu/OPD.
Mzdata [20]	1.05	2005	HUPO Proteomics Standards Initiative.	To capture peak list information. Its aim is to unite the large number of current formats into one.	
AGML [21]	2.0	2004	Medical University of South Carolina.	To model the concept of annotated gel (AG) for delivery and management of 2D Gel electrophoresis results.	AGML Central http://bioinformatics.musc.edu/agml2/web/pages/
ProtXML [22]	n/a	n/a	Institute for Systems Biology	A format for storage, exchange, and processing of protein identifications created from ms/ms-derived peptide sequence data.	
PepXML [22]	n/a	n/a	Institute for Systems Biology	A format for storage, exchange, and processing of peptide sequences derrived from ms/ms scans.	

Table I
AVAILABLE STANDARDS, CREATORS AND AVAILABILITY.

shortcomings that had hindered the implementation of the standard.

Due to the nature of the field, the community has realized that there will exist a plethora of competing formats and a number of specifications on minimum information required within different fields has been devised, e.g. MIAME [25] (Minimum Information About a Micro-array Experiment) for micro-array data, MIAPE [26] (Minimum Information About a Proteomics Experiment) for proteomics data, MIGS (Minimum Information about a Genomic Sequence) for genomics data and MIRIAM [27] (Minimum Information Requested In the Annotation of bio-chemical Models) for system biology models. They often require use of controlled vocabularies. Other requirements could be literature source references or information about from which organism data was collected.

Given the situation today there will continue to exist a large number of XML-based bioinformatics formats in the future. In addition, several formats for storing the same type of data and different versions of the same formats will be used simultaneously.

III. Goals and Motivation

There are a number of tools available for management and processing of XML data. In addition, there are also a number of dedicated tools available for handling data in the special designed standards for bioinformatics. Examples of such tools are simulation tools and visualization tools, e.g. Cytoscape [28] and GNU MCSim [29], that offers import and export in various predefined XML formats. The focus for this paper are applications where there is a need for complex information retrieval, i.e. where the user needs to combine the data to gain new information. In addition we assume that the user is interested in data from several different databases, exported in several of the standards described above. The most natural way to provide this is to store the data within a database and query it.

However, for bioinformatics this puts hard requirements on the data management solution. On the one hand the data that we want to use is downloaded from the web in one of the many XML standards that are available within the area. This means that we need solutions where it is fast to import the data into the database and where little effort needs to be spent on designing the storage solution. On the other hand many of the tasks that we are interested in, for instance, combining and comparing information from several datasets or graph analysis, requires quite complex queries on the dataset. Previous studies have shown that native XML solutions do not perform well when the query complexity grows [30].

The main goal for this work is to explore ways for more efficient data processing within bioinformatics. Our primary goal is query efficiency, easy import and reuse of data in any of the bioinformatics standardized formats is also an important issue. We will address these issues in two important tools. The first addresses graph processing capabilities, and suggests a standard independent extension to XQuery that provides easy to use and efficient graph processing of XML data. The second tool provides an easy way of exploring more efficient storage models for the data. The motivation for this is that a pure native XML storage yields too inefficient querying for the data while a relational storage provides more efficient querying. The goal for our second tool is to provide easy creation and import of data to a hybrid XML storage model.

For our first tool we address cases where the databases provide data export in one or several XML exchange formats for protein interactions, e.g. SBML [9], PSI MI [2], and CellML [15]. These datasets available in XML provide descriptions of interaction networks or graphs [31]. Therefore, it would be beneficial for the user to enable querying and analysis based on the XML format, i.e. to be able to query the data using XQuery. Our goal is to find a solution that can preserve the full functionality of XQuery and in parallel provide an efficient handle for graph analysis. As many standardized data representation formats exist for the area it is important to find a general solution where all XML-based data formats can be used.

To reach our goal we need a way to enable graph processing directly in the XML environment. One solution would be to implement graph queries directly in XQuery [32]. However, our initial studies of this [30] were disappointing. The queries get complex and inefficient to compute, which make it impractical for biologists that may have limited knowledge in programming. Therefore, we want to provide graph functionality within XQuery by extending the language. As we do not want to change the core functionality of XQuery we want to add graph functionality through addition of built-in functions which make them available directly from XQuery. The first tool we describe presents an extension to XQuery which allows extended analysis on graphs. The main application for the work is biological interactions, but the extension is generic and capable of handling graphs represented as XML also for other applications. In section IV we give a general description of the chosen solution, our implementation and an evaluation of the tool.

For our second tool we will investigate how well hybrid databases as provided by modern relational database managers [33], [34], [35], [36], [37] can match the requirements of bioinformatics. With hybrid solutions the user can choose to use either native or relational storage for his data. It is also possible to combine the solutions and store parts of a document as XML and other parts of it as relations. Consequently the user can work with XQuery for parts of the data and SQL for other parts. He can also choose to retrieve results from queries in the format the data is stored or to convert it to the format he prefers.

We aim at combining the benefit of native XML databases, which is an easy to use solution, with the efficiency provided by relational databases. The main drawback with this solution is the cost for designing the hybrid storage, i.e. to decide which parts of the XML code that should be stored as relations and which parts should preferably be preserved in its original XML structure. The work by Moro et al. [38] addresses this problem by providing guidelines for when parts of the XML structure should be translated or not. They also provide a tool where the user can design hybrid XML storage.

In our case the problem is a bit different. Our starting point is the already available standards within bioinformatics which provides us with the XML data model and in most cases also the XML schema defining this model. Therefore we want a solution where we can use this information as a basis for the hybrid storage. We have chosen to adapt the solution by Amer-Yahia et al. [39]. They present a system, ShreX, which automatically can map an XML schema to relational tables and import the resulting schema and dataset into a database.

In section V we present a tool HShreX which extends the original system to hybrid databases. We have also extended it with some further functionality to vary the mapping into relations. We present the main architecture behind the system, an illustrating example and an evaluation illustrating the benefit of using the system.

Together the two tools provide a powerful workbench for analyzing bioinformatics data. In practice they can be used as two separate tools. However, at the end of the article we discuss how they can be combined into one single environment.

IV. An XQuery extension for graphs

Our first tool provides efficient and easy to use graph analysis functionality for XQuery and was previously presented in [1]. In particular, we want to find a solution that is applicable to all standards within the area of molecular interactions and pathway data. We also want to find a solution where existing efficient graph algorithms can be reused within the environment. We choose to do this by extending XQuery with specialized functions for graph analysis. The goal for our work is to find a solution that adds new graph functionality that blends well into existing XQuery functionality and does not introduce new features to XQuery itself. At the same time we want the data, algorithms and results to be accessible from XQuery. As the solution should be independent of XML format, graphs should be freely modeled by the XQuery/XPath expressions and changes to the original XML data should not be necessary.

A. Architecture

One of the challenges is to provide a solution that is independent of XML format, as the external functions must know which parts of the XML file constitutes the graph. To deal with this we define a common graph model that the supplied functions are operating on. In addition to this our solution must contain handles for connecting the original XML representation to the general graph model. The selected graph model enables labeled directed graphs. It has been chosen so that it captures the most common properties for biological pathways.

Definition A *graph* is defined as a quadruple $G := (V, E, FV, FE)$ with:

- V, the set of vertices.
- E, the set of edges. An edge describes the relation between its two endpoints - the two connected vertices $((v, w) \in E; v, w \in V)$. Furthermore, parallel edges are not allowed, so no two distinct edges may have the same endpoints. Edges with identical endpoints, so called loops, are not allowed.
- I, is a set of identifiers used to denote properties, e.g. name or weight of edges.
- L, is a set of labels, i.e. the values of the properties, usually a substance name or the weight of an edge.
- $FV : V \times I \rightarrow L$, is a set of mappings associating labels for each vertex and a given identifier.
- $FE : E \times I \rightarrow L$, a set of mappings associating labels for each edge and a given identifier.

Hence, labels can be attached to vertices and edges to provide additional information, for instance, enabling graph algorithms to incorporate weights. Graphs may be directed or undirected. The focus for this work have been to investigate connectivity. Therefore we made the restriction to not allow parallel and looping edges since they not give extended information to the graph queries of our interest. The resulting model can capture all information inherent in the protein interaction and pathway standard descriptions presented in the previous sections. Hyperedges, i.e. edges connecting several nodes, can be represented by a set of edges in our model while identifiers and labels can be used to represent information not directly captured by the vertices and edges.

The final step needed for our solution is a way to map the data between the original XML format and our graph model. To achieve the required functionality we need handles to load, get and execute graph analysis on our graph model. The load functionality constitute mapping from the original XML data to the graph model. The mapping between the original XML format and graph model is done by specifying XPath expressions. These define which parts in the original format that corresponds to an edge, vertex or a label. Executing these expressions will result in pointers to XML items that are used to build the desired graph. The remaining functionality is used to import graph data back to the XQuery environment. This can be done either by fetching all or part of the graph (*get*) or by retrieving a graph as part of

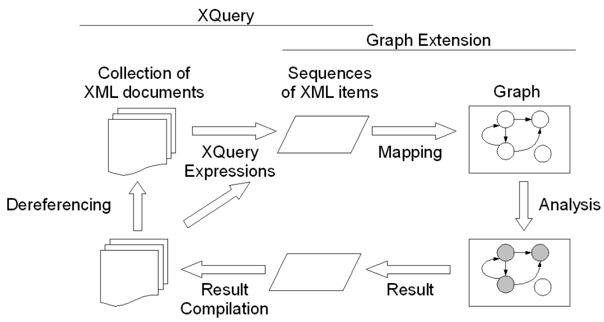


Figure 3. Architecture for the extension

executing an supplied algorithm on the graph (*execute*). As the data returned from the graph package normally constitute only part of the data (the graph information or in most cases a subgraph) from the original format we decided to use GraphML for its representation instead of the original graph format. This gives a clear distinction between returned results and the original data file.

The resulting architecture for our extension is depicted in Figure 3. As desired XQuery is used to address the data subject to analysis; the graph extension uses the graph model to process the analysis. The user can create graphs that are represented internally in the graph extension. Other XQuery expressions allow the user to execute graph queries by utilizing functionality available in the external graph package. The result from these queries is received by the user as an XML representation of the graph. If the user wants he or she can then link these results to the original XML file by a referencing mechanism.

B. Implementation

A prototype implementation in Java was built on the native XML database eXist, its XQuery processors and the JUNG graph framework in order to investigate usability, performance and overall strength and weaknesses in practice. We chose eXist version 1.0.2 (*exist.sourceforge.net*) since it is an open source native XML database with an extensible XQuery implementation in Java. The JUNG graph framework version 1.7.6 (*jung.sourceforge.net*) has been chosen to implement the graph model. JUNG is like eXist written in Java and supports directed and undirected graphs, hypergraphs, bipartite graphs and labels for vertices and edges; therefore it easily satisfies the proposed graph model.

To enable an environment where it was easy to experiment with different functionality and several graphs in parallel we introduced a set of functions. First we added two functions to create and delete graphs explicitly (*createGraphs* and *releaseGraphs*). Secondly the load functionality is implemented by a set of easy to use functions to define the properties like vertex, edge and their labels (*loadVertices*, *loadLabeledVertices*, *loadEdges* and *loadLabeledEdges*). Finally we implemented two functions for retrieveing the graph data or results of an algorithm (*getGraphs* and *execute*). In this implementation, especially the load functions rely on related sequences. Therefore, the document order, i.e. the order in which XML nodes appear in the XML serialization of a document, is the default order if no ordering is defined.

The required reference mechanism that are used to link from graph data back into the original XML document are implemented according to Chamberlin et al. [40] by two functions, *fn:ref* and *fn:deref*. Obviously, the functions can work correctly only if the node IDs are stable, regardless of changes to the document if updates are allowed. Updates to XML documents are not considered in the graph extension.

C. Example

We illustrate how the extension works by showing an example using the SBML [9] data. An example data model in SBML is given in figure 4. The example in figure 5 illustrates the usage of the functions in the implementation.

- The root element of the XML data is bound to `$doc`. (Expression 2)
- One directed graph is created and bound to `$graph`. (Expression 3)
- Two variables bound the IDs of interesting molecules. (Expression 4 and 5)
- Then all IDs of the species element are selected by an XPath expression, loaded as vertices into the referenced graph. (Expression 6)
- A FOR-expression is used to access and load each reaction into the graph. The URI of each reaction serves as edge ID and is retrieved with `xqueryp:ref($reaction)`. The expression `$reaction/s:listOfReactants/s:species-Reference/@species` relates to the vertices defined in the previous step. (Expression 7)
- After defining the graph's properties the shortest path is calculated, the returned XML node is in GraphML format, the edge ID holds the reference to the original SBML data. The edge IDs are selected by `//edge/@id` and then resolved by `xqueryp:deref` which are the reactions in SBML representing the shortest path between the specified substances. (Expression 8)
- Finally, the graph is deleted. (Expression 9)

D. Evaluation

To evaluate our approach we have performed a series of experiments. We were in particular interested in three properties; the overall performance for graph analysis for biological pathways data; comparing this with using plain XQuery; and finally an analysis of the performance of loading graphs. All experiments were done on a notebook

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1">
  <model id="Tyson1991CellModel_6" name="Tyson1991_CellCycle_6var">
    + <annotation>
    <listOfSpecies>
      + <species id="C2" name="cdc2k" compartment="cell">
      + <species id="M" name="p-cyclin_cdc2" compartment="cell">
      + <species id="YP" name="p-cyclin" compartment="cell">
      ... more species
    </listOfSpecies>
    <listOfReactions>
      <reaction id="Reaction1" name="cyclin_cdc2k dissociation">
        <annotation>
           <rdf:li rdf:resource="http://www.reactome.org/#REACT_6308"/>
           <rdf:li rdf:resource="http://www.geneontology.org/#GO:0000079"/>
        </annotation>
        <listOfReactants> <speciesReference species="M"/> </listOfReactants>
        <listOfProducts>  <speciesReference species="C2"/>
                          <speciesReference species="YP"/>  </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply> <times/> <ci> k6 </ci> <ci> M </ci> </apply></math>
          <listOfParameters> <parameter id="k6" value="1"/> </listOfParameters>
        </kineticLaw>
      </reaction>
      + <reaction id="Reaction2" name="cdc2k phosphorylation">
      ... more reactions
    </listOfReactions>
  </model>
</sbml>
```

Figure 4. SBML representation of the Tyson Cell model as it is represented in the Biomodels (www.biomodels.net) database. The example has been abbreviated and simplified to improve readability.

```
1: declare namespace s = "http://www.sbml.org/sbml/level2";
2: declare variable $doc {doc("reactome/homo_sapiens.xml")};
3: declare variable $graph {graph:createGraphs("org.exist.xquery.modules.graph.JUNGGraphImpl",true())};
4: declare variable $source {"R_111584_xanthosine_5_monophosphate"};
5: declare variable $target {"R_29398_Pyruvate"};
6: graph:loadVertices($doc//s:listOfSpecies/s:species/@id, $graph),
7: for $reaction in $doc//s:listOfReactions/s:reaction
     return   graph:loadHyperEdge(xqueryp:ref($reaction),
                         $reaction/s:listOfReactants/s:speciesReference/@species,
                         $reaction/s:listOfProducts/s:speciesReference/@species,$graph),
8: xqueryp:deref(graph:execute("dijkstraShortestPath",($source, $target), false(),
                                                         true(), $graph)//edge/@id),
9: graph:releaseGraphs($graph)
```

Figure 5. Example on how to use extended graph functionality in XQuery.

with Windows XP Professional, a 1.6GHz Pentium Mobile and 1GB main memory.

The first experiments exemplify how well the graph extension scales for graphs with a few thousand vertices. In the experiments sample test series were successfully and efficiently executed on real application data from the Reactome [6] and KEGG [7] databases. The Reactome data set is stored in one SBML document of 1.2MB comprising 3054 substances and 1917 reactions which were resolved into 4832 edges. The KEGG data set is stored in 92 SBML files with a total of 1,2MB comprising 1652 substances and 1122 reactions which were resolved into 1296 edges.

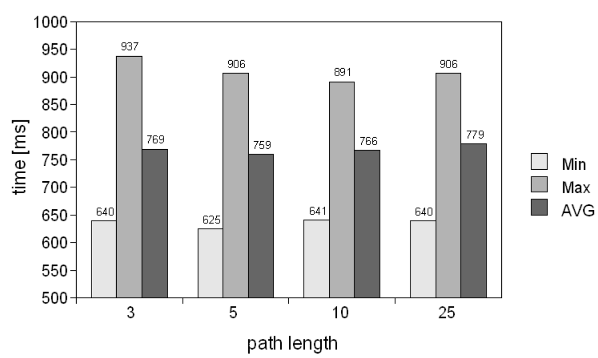


Figure 6. Performance on the Reactome dataset.

Figure 6 shows the results of 100 passes of the Dijkstra shortest path on pairs of substances from the Reactome data set where the path length was 3, 5, 10 and 25, i.e. the query presented in section 4 with selected start and end nodes. The reason for running each query 100 times is to reduce the impact of other processes, such as Java garbage collection that may affect the result. Analogously, figure 7 shows the results of 100 passes of the Dijkstra shortest path on pairs of substances from the KEGG human data subset where the path length was 3, 5, 10 and 14 on the same query.

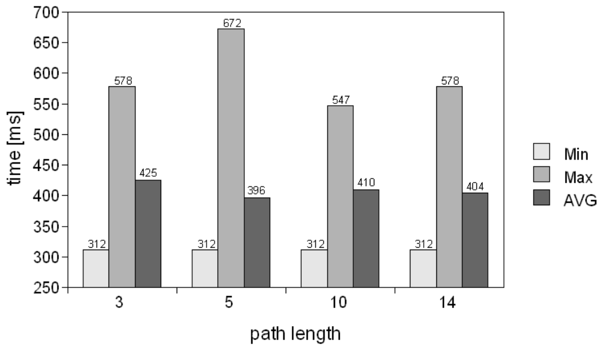


Figure 7. Performance on the KEGG dataset.

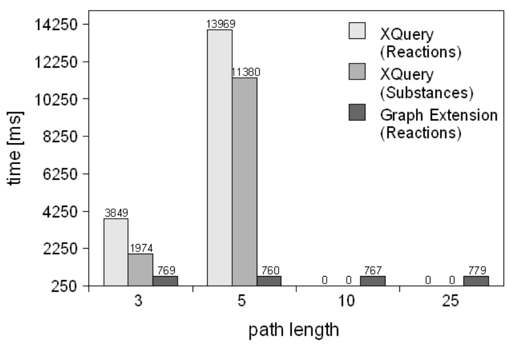


Figure 8. Performance comparison of our extension with plain XQuery.

The query times include creating, populating and deleting the graph, the execution of the algorithm and the XML representation of the paths based on the original SBML data. For these tests caching of shortest path results within JUNG was deactivated. This shows that our graph extension works well for the tested data.

Secondly, we wanted to compare our results with using plain XQuery. An adapted depth-first search algorithm in XQuery was implemented for comparison. All shortest paths are searched within the given depth. The implementation shown in figure 9 uses two functions. The *local:findPath* creates the spanning search tree recursively, found paths are marked with a `<found/>` element. The function expects three parameters, the cut-off depth, the start and the end vertex of the path. The second function, *local:getParent* is used to traverse the found path up and collect the parents. This implementation demonstrates how complicated and inefficient it is to build an algorithm based on sequences of items and temporary XML fragments.

Figure 8 illustrates the results together with the query times on the same data set with the graph extension. The query for XQuery (Reactions) takes longer because it presents the path with all reaction elements, whereas the query for XQuery (Substances) presents a condensed version as a list of elements with the reaction IDs and only the substance IDs found by the path search as attributes. It must be noted, that the comparison with the graph extension is not completely fair. The Dijkstra's shortest path algorithms used within the graph extension only returns the single shortest path whereas the XQuery implementation completely explores all shortest path within the specified depth. The query times for a path length of 5 are still acceptable if the data volume is disregarded, but the query does not finish on the same data set within an hour with 10 as cut-off depth. One reason is certainly that with every step the search tree grows tremendously by the fact that the query does not sufficiently detect cycles.

Finally, we wanted to analyze the performance of loading graphs into the graph module to understand how much of the total execution time that were spent on creating the graphs. For these experiments we used the Reactome dataset. As for the total execution time we compared our loading performance with an XQuery expression retrieving the same information from the data file. From this experiment we can conclude that loading the data is very fast. In fact, most time is spent on retrieving the data from the XML file. The execution of Dijkstra's shortest path is even faster and because of this the differences between different path lengths are marginal. The divergence between different path lengths is roughly between 5ms and 20ms on average. In comparison, the difference between minimum and maximum performance are significant, but still under half a second.

A final remark is that the presented results refer to small amounts of data in particular in regards to data volumes databases are built for and for our tests in memory processing could be used. Query times increase dramatically if the whole KEGG data set is utilized including different species (132MB, 12122 files), because data is stored highly redundantly. In that case most time is spent on the XQuery expressions to retrieve the sequences of items to map onto vertices and edges. The data volume to process the analyze is reduced because of the integrated duplicate elimination. This behaviour is beneficiary for scenarios, where we can expect that the user loads the data into the database and then runs a series of analysis on the dataset.

E. Discussion

The general architecture for our extension proposes that XPath expressions are used to declare the XML data and the graph model. In our implementation we choose to implement this as a set of load functions which makes use of side effects. This is controversial since XQuery is a side-effect free query language. The main problem with introducing side effects is that query optimization is hindered. The order of execution of the graph functions matters putting restrictions on optimization. However, the evaluation for all other XQuery expression can still be optimized without further limitations. Our view is supported by Chamberlin et al. [40] who state that global optimization is difficult in a mixed language environment.

```
declare function local:findPath($start as xs:string,
                                $end as xs:string, $n as xs:integer) {
  for $species in
    $doc//s:reaction[s:listOfReactants/s:speciesReference/@species = $start]
                               /s:listOfProducts/s:speciesReference/@species
  return<item reaction="{$species/../../../@id}"species="{$species}">  {
    if($species = $end) then <found/> else
      if($species = $start or $n = 1) then () (: loop or max :)
       else local:findPath($species, $end, $n - 1)} </item>};

declare function local:getParent($itema as node()?) {
  if($itema[@species]) then (local:getParent($itema/..), <node>
     {$itema/@species} {$itema/@reaction} </node>)         else ()};

<paths>{for $found in local:findPath($source, $target, $maxLength)//found/..
   return <path> <node species="{$source}"/{local:getParent($found)} </path>}
</paths>
```

Figure 9. XQuery version of findpaths used for comparison with our extension.

To avoid side effects one solution would be to implement the graph as an extended index to the database. This is possible in for instance eXist 1.1. With this solution the functionality for creating the graph would be analogous to creating an index and performed when new files are loaded into the database. We did not choose this solution, as it would give us less freedom to experiment with different graph realizations of a dataset which was a goal for this version of the tool. An alternate solution would be to make use of further developments of XQuery like XQueryP [40], [41] and XQuery! [42]. XQuery! proposes to extend XQuery with a set of side-effecting operations, especially handy for XML updates [42]. Therefore it introduces a new operator that allows applying a sequential mode to an XQuery fragment. XQueryP introduces even more features to extend XQuery for application logic [40], [41]. Another approach to separate the concerns of assembling the graph and querying it using XQuery could be to annotate the XML schema of the source format defining the desired structure and elements of the graph.

V. User designed hybrid storage

Our second tool [43] investigates how well hybrid databases as provided by modern relational database managers [33], [34], [35], [36], [37] can match the requirements of bioinformatics. With hybrid solutions it is also possible to combine the solutions and store parts of a document as XML and other parts of it as relations. Our aim is to combine benefit of native XML databases, which is an easy to use solution, with the efficiency provided by relational databases and minimizing the cost for designing the hybrid storage. Our starting point is the already available XML schema defining the model for the chosen standard. Our tool allows the user to take benefit from and experiment with hybrid XML storage.

A. Architecture

HShreX [43] is a tool that automatically can, from an XML Schema, create a native, relational, or hybrid data model. HShreX builds upon a previous tool ShreX [39] developed for shredding XML data into pure relational storage.

The starting point for HShreX is the XML Schema. When the user loads a schema in HShreX, it first creates an internal schema model, which is a tree-like structure specifying the details of the schema. Once the schema model has been created it is traversed in order to determine mapping information (e.g., the simple XML element *name* should be mapped to a field called *name* in the table *xyz*), from which a relational model (that can be pure relational, native, or hybrid) is created. The exact characteristics of the resulting model depend on a default set of shredding rules which can be influenced by using annotations in the XML schema. The user can now inspect the relational model and redesign it using schema annotations until a desired one has been created. When a satisfactory model has been created, it can be loaded onto a live database. This is done by a relation generator which generates scripts adapted to the chosen relational database manager. After this step, data can be loaded by opening XML files. A data converter looks up mapping information and generates a script with tuple insertion statements and runs it when all the data has been read. The architecture is visualized in figure 10. The default shredding rules include the following behavior:

- Complex elements are shredded into tables. All tables will get a primary key field named *shrex_id*. If the complex element is not a root element it will also get a foreign key field named *shrex_pid* that point to its parent. This preserves the tree structure in the original XML data. If the complex element can have simple content (i.e., text content), a special field is created in the table to hold any such content.
- Simple elements are shredded into columns in their parent table if they can occur at most once under their parent. If a simple element can occur more than once under its parent it will be outlined to a separate table.
- Attributes are shredded into columns in their parent table.

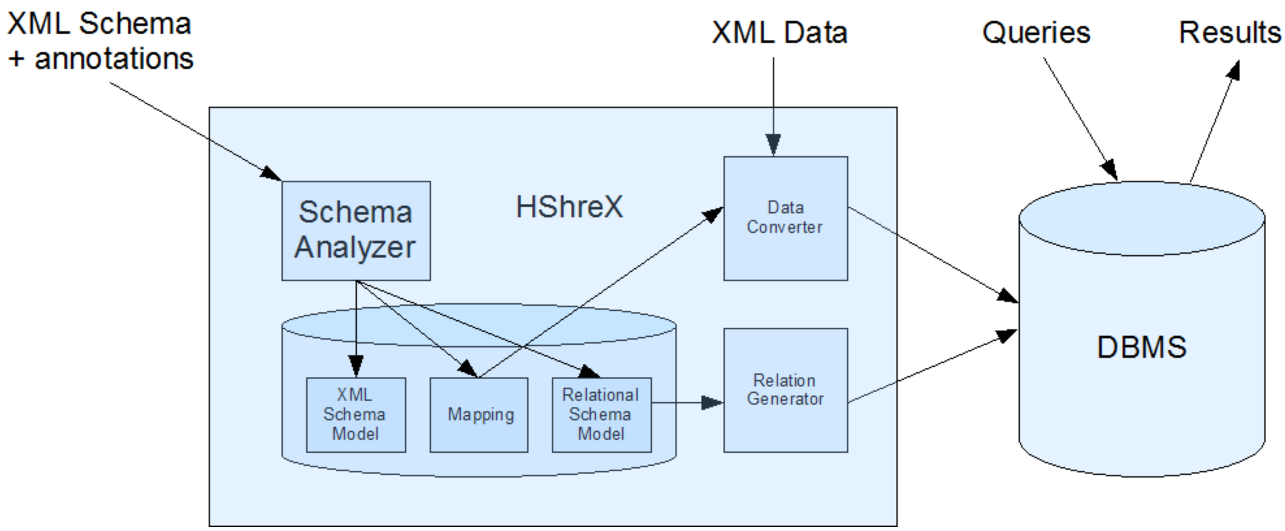


Figure 10. HShreX architecture

```
<xs:element name="minisbml">
  <xs:complexType>
     <xs:sequence>
        <xs:element name="author" type="PersonType"/>
        <xs:element name="molecule" type="Moleculetype"
                    minOccurs="1" maxOccurs="unbounded"/>
        <xs:element name="reaction" type="Reactiontype"
                    minOccurs="0" maxOccurs="unbounded"/>
     </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="Moleculetype">
  <xs:attribute name="name" type="xs:string"
                use="required"/>
</xs:complexType>

<xs:complexType name="Reactiontype">
  <xs:sequence>
     <xs:element name="reactant" type="Moleculetype"
                 minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="product" type="Moleculetype"
                 minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string"
                use="required"/>
</xs:complexType>

<xs:complexType name="PersonType">
  <xs:sequence>
     <xs:element name="name" type="xs:string"/>
     <xs:element name="affiliation" type="xs:string"
                 minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Figure 11. Sample XML schema

In figure 11 and 12 a sample XML schema is shown with an accompanying XML document, respectively. The schema lacks annotations so it will be processed by HShreX using the default shredding rules, yielding the relational model found in figure 13.

Shredding a schema using just the default rules will in most cases create a pure relational model. The only exception is elements that have the type *anyType*, i.e. elements that have no XML structure definition in the schema, which are mapped to XML. In many cases this will cause a large number of tables to be created, which can be a problem because it makes the model hard to understand and overview. Another problem with models that suffer from an explosion of tables is that semantically related data run a risk of being separated into different tables. Combined this can make the task of writing queries complex and performance can suffer. Therefore, HShreX allows the default shredding rules to be influenced via annotations. A number of annotations are supported and they are used on the schema to change the default shredding rules. A document describing all annotations supported by HShreX can be found on http://hshrex.sourceforge.net/. Here follows a few of the more important annotations:

```
<minisbml>
  <author>
     <name>Lena Strömbäck</name>
     <affiliation>IDA</affiliation>
  </author>
  <molecule name=M1/>
  <molecule name=M2/>
  <molecule name=M3/>
  <reaction name=R1>
     <reactant name=M1/>
     <reactant name=M2/>
     <product name=M3/>
  </reaction>
  <reaction name=R1>
     <reactant name=M3/>
     <product name=M2/>
  </reaction>
</minisbml>
```

Figure 12. Sample XML document

- *maptoxml* – makes this part of the XML tree to be stored natively. The annotation can be used on both complex and simple elements.
- *ignore* – this part of the XML tree will be ignored, i.e.

minisbml:

shrex_id
1

minisbml_molecule:

shrex_id	shrex_pid	name
1	1	M1
2	1	M2
3	1	M3

minisbml_reaction:

shrex_id	shrex_pid	name
1	1	R1
2	1	R2

minisbml_reaction_reactant:

shrex_id	shrex_pid	name
1	1	M1
2	1	M2
3	2	M3

minisbml_reaction_product:

shrex_id	shrex_pid	name
1	1	M3
2	2	M2

minisbml_author:

shrex_id	shrex_pid	name
1	1	Lena Strömbäck

minisbml_affiliation:

shrex_id	shrex_pid	affiliation
1	1	IDA

Figure 13. Generated relational tables.

it will not be represented in the resulting data model.

- *outline* – used on simple elements (or attributes) where it is desired that they be stored in a separate table.
- *withparenttable* – used to merge a child with its parent in order to reduce the number of tables in model. This annotation can be used only for children with a single occurrence in the parent.
- *tablename* – can be used to simply rename a table but a more powerful use is to merge two tables that do not have a parent/child relationship (in those cases the annotation described above, *withparenttable*, is used).

Maptoxml, *ignore* and *withparenttable* are new annotations for HShreX whereas the other annotations work as in the previous ShreX tool. In addition the system allows varying the underlying basic shredding principle. This will not be further discussed here.

B. Implementation

HShreX is developed in Eclipse and written in Java version 1.6. The main development platform is Windows Vista, but HShreX also runs on Windows XP and Linux. A large part of what HShreX does is processing XML and for that Xerces2-J version 2.9.1 is used. HShreX knows how to communicate with IBM DB2 9.5 fixpack 1 or later and Microsoft SQL Server 2008 but in order to do that HShreX needs drivers supplied by the vendors. For Microsoft SQL Server sqljdbc4.jar is used and for IBM DB2 the dependency is db2jcc4.jar. A large set of unit tests is part of the HShreX sourcebase and to run them one needs JUnit version 4.3 or later. HShrex together with documentation can be obtained in binary and source form at http://hshrex.sourceforge.net/.

```
<xs:element name="minisbml">
  <xs:complexType>
     <xs:sequence>
        <xs:element name="author" type="PersonType"
                    shrex:maptoxml="true"/>
        <xs:element name="molecule" type="Moleculetype"
                    minOccurs="1" maxOccurs="unbounded"
                    shrex:ignore="true"/>
        <xs:element name="reaction" type="Reactiontype"
                    minOccurs="0" maxOccurs="unbounded"
                    shrex:maptoxml="true"/>
     </xs:sequence>
  </xs:complexType>
</xs:element>
```

minisbml:

shrex_id	author
1	<author> <name>Lena Strömbäck</name> <affiliation>IDA</affiliation> </author>

minisbml_reaction:

shrex_id	shrex_pid	xml
1	2	<reaction name=”R1”> <reactant name=”M1”/> <reactant name=”M2”/> <product name=”M3”/> </reaction> <reaction name=”R1”> <reactant name=”M3”/> <product name=”M2”/> </reaction>
2	1	<reaction name=”R1”> <reactant name=”M3”/> <product name=”M2”/> </reaction>

Figure 14. Hybrid mappings with maptoxml and ignore.

C. Example

To illustrate how HShreX can be used we give two examples of using the annotations to design the shredding. The first example in figure 14 illustrates how the hybrid mapping can be used. In this example the aim is to map the information about authors and reactions to XML and remove information about molecules (assuming these are not interesting for the current information need). This kind of mapping is common in bioinformatics since most of the bioinformatics standards are very rich and define a large amount of elements for representing various portions of information. In many real cases parts of this information are not interesting for the end user or many of those elements is not even used by the source exporting the data.

To achieve this shredding we have added *maptoxml* anno-

```
<xs:element name="minisbml">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="author" type="PersonType"
                  shrex:withparenttable="true"/>
      ...rest of definition in figure 2...
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:complexType name="Reactiontype">
  <xs:sequence>
    <xs:element name="reactant" type="Moleculetype"
                minOccurs="0" maxOccurs="unbounded"
                shrex:tablename="participant"/>
    <xs:element name="product" type="Moleculetype"
                minOccurs="0" maxOccurs="unbounded"
                shrex:tablename="participant"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>
```

minisbml:

shrex_id	name
1	Lena Strömbäck

minisbml_molecule:

shrex_id	shrex_pid	name
Content as in Figure 4		

minisbml_reaction:

shrex_id	shrex_pid	name
Content as in Figure 4		

participant:

shrex_id	shrex_pid	name
1	1	M1
2	1	M2
3	1	M3
4	2	M4
5	2	M5

minisbml_author_affiliation:

shrex_id	shrex_pid	name
Content as in Figure 4		

Figure 15. Example of withparenttable and tablename.

tations to the *author* and *reaction* elements in the definition of *minisbml*. As shown in the bottom of the figure this results in adding *author* as an attribute in the *minisbml* table. The *minisbml_reaction* will still be generated, but with all its content as XML in the XML column of the table. To remove the molecule information we have added the annotation *ignore* in the XML schema.

Figure 15 demonstrates an alternative way to make the relational mapping easier to understand and use. In this case we do not want to use hybrid storage. Instead the goal is to remove unnecessary relations in the generated shreddings; in this case we can move the *author* to the *minisbml* table since only one author per table is allowed, and force all participants of reactions to be shredded into on single relation, thus decrease the number of relations generated by HShreX.

Removing the *author* relation is achieved by using the annotation *withparenttable*. To shred several substructures into the same table the annotation *tablename* can be used as renaming substructures into the same *tablename* forces the corresponding data to be shredded into the same table.

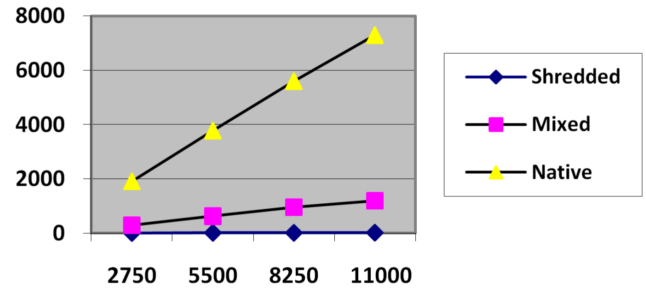


Figure 16. Query performance [ms] with growing datasets (number of UniProt entries on the y axis).

```
SELECT accession
FROM entry, accession, comment, subcellularLocation, location
 WHERE entry.shrex_id = accession.shrex_pid
 AND entry.shrex_id = comment.shrex_pid
 AND comment.shrex_id = subcellularLocation.shrex_pid
 AND subcellularLocation.shrex_id = location.shrex_pid
 AND location.nodeValue='Cytoplasm';
```

Figure 17. UniProt query (for mapping 1)

D. Evaluation

In this section we will evaluate the benefit of working with HShreX. There are two issues, performance of queries and the complexity of data models. We have chosen to work on data available for two commonly used bioinformatics standards SBML 2.1 [9] and UniProt [17]. All tests are done on an AMD Athlon Dual Core 2.9 GHz and 4 GiB RAM.

For our first test we have designed three different data models. The first one is a pure native representation where the XML data files are stored as XML in an XML attribute in one main relation. The second one is a mixed representation, where we have translated parts of the XML into relations and kept other parts as XML. The intuition for creating the mixed representation is to create a hybrid data model reflecting the semantics of the original SBML standard. The third data model is the purely shredded representation produced without any annotations.

There is a clear relation between the choice of model and the query performance as illustrated in figure 16. The query (as it is formulated in SQL for the purely shredded mapping) is listed in 17. The example illustrates the benefit of using the mixed representation in a case where we are joining many tuples. In this case we want to combine data from UniProt (www.uniprot.org). Here, the native representation results in poor performance, while the shredded version is very fast. However, the mixed representation gives a considerable improvement over the purely native representation. This shows that shredding parts of the XML data could have a considerable improvement of the performance.

To illustrate the complexity of the created models we

	Data model	Native	Mixed	Shredded
SBML	*Nr of annotations*	1	21	0
	Nr of relations	3	8	121
UniProt	*Nr of annotations*	1	24	0
	Nr of relations	2	32	121

Table II
INFORMATION ABOUT THE DIFFERENT DATA MODELS

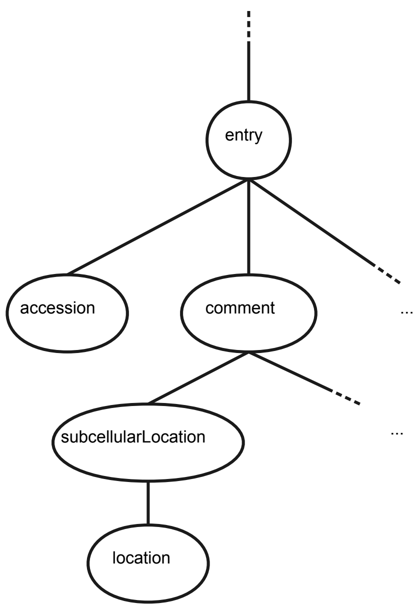


Figure 18. Part of the UniProtKB XML tree. The *comment* node can be seen in figure 2.

present more details of selected models for SBML and UniProt. Table II presents the number of annotations needed and the number of produced relational tables for these mappings. The purely native and the mixed representation produce data models with a limited amount of tables, while the purely shredded model generates many relational tables. This explosion of tables causes data that semantically belong together to be shredded into many places in the data model. The mixed version of our data models creates a data model that provides relational storage for entities that we assume will be commonly accessed in queries and native XML representation used for other parts. We do not use the ignore annotation for this example to make the three models comparable in information content. The examples in the previous section illustrate the intuition on how to build this mapping, basically we add *maptoxml* annotations to the parts to be stored as native XML and *withparenttable* annotations to levels in the XML-tree that we want to omit. As shown in table 2 this is easily done and we only need around 20 annotations for the given schemas.

To further illustrate the impact of shreddings we have also evaluated query performance for all possible hybrid representations relevant for the query in figure 17. There

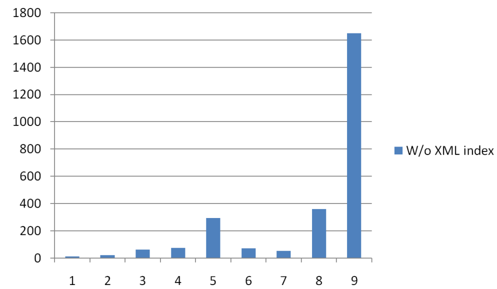


Figure 19. Query performance [ms] for shredded mapping (1), different hybrid mappings (2–8) and native mapping (9).

are seven possible hybrid mappings where a varying degree of the XML subtree affected by the query is stored as a XML value instead of being shredded to relations. Since the sub-tree has two branches (see figure 18) we can design eight different hybrid mappings; *accession* (2 in figure 19), one of *location* (3), *subcellularLocation* (4) and *comment* (5) or *accession* together with one of *location* (6), *subcellularLocation* (7) or *comment* (8) can be mapped as XML. The mappings where *comment* is shredded but *accession* and/or *subcellularLocation* or *location* is mapped as XML all run in under 75 ms on the test system with a dataset of 2750 entries and using no XML indexes. When *comment* is mapped as XML the run time rises to 300–360 ms (depending on whether *accession* is shredded or not). The native mapping is much slower (1650 ms). This demonstrates how choosing a preferable shredding gives acceptable performance and a more comprehensible table structure than the purely shredded mapping. Data stored as XML values do not need serializing back into XML which is a time benefit for certain types of data. Which mapping results in the best query performance while keeping a comprehensible structure is non-intuitive. How efficient a mapping is in terms of performance depends on the query, the structure of the schema and distribution of data within the structure.

VI. RELATED WORK

Regarding related work there is a lot of work on extended functionality for XQuery. Here, the Mark Logic Corporation provides for its XQuery implementation several function libraries to ease application development [44]. In addition, the eXist community has added a number of new functions as function modules to the XQuery implementation, for example a mail, math, SQL and spatial module. Our XQuery extension combines the ideas above to realize graph processing based on the additionally introduced graph model. We also looked at relational database systems and found similar tendencies. Besides, for spatial data applications relational database vendors recognized the need of graph support in areas like biology. One example is the Life Science Platform by Oracle [45]; another is the Systems Biology Graph Extender (SBGE) for IBM's DB2 database system [46].

The SBGE is of particular interest since it resembles our graph extension tool. It introduces a data model which introduces graphs as a first-class SQL data type. This means that graphs can be manipulated the same way as other data types. In addition it defines operators that can convert data between the graph representation and relational tables containing the corresponding information. The workflow of this extension is very similar to the one of our graph extension. Analogue to the load-functions data stored as SQL tables can be converted to the graph representation. Then operations can be efficiently performed on the graph representation of the data as can be done with our execute-function. Finally, the results can be stored as plain SQL tables or SQL tables containing graphs, similar to access the graph data through the get-function or the XML node returned by the execute function. Similar to our current implementation, the SBGE implementation requires that each graph fit into main memory. SBGE functions can be seamlessly composed in a single SQL query with user defined functions (UDF) written in Java.

The HShreX tool, on the other hand, combines ideas from two related areas for XML storage. The first is the work on automatic shredding of XML documents into relational databases by capturing the XML structure or based on the DTD or XML schema for the XML data [39], [47], [48]. The intention with these approaches is to create an efficient storage for the XML data. The resulting data model is often not easy to understand and is usually hidden from the user via an interface providing automatic query translation of XQuery into the model. Several authors also explore the efficincy of strategies for shredding XML to relational engines [49], [50], [51].

The other related area is work on hybrid XML storage, as provided by the major relational database vendors. The underlying representation for the XML type differs, in some cases it is a byte representation of the XML whereas in other cases it is some kind of shredding of the XML data [33], [34], [37]. These database vendors provide a number of tools to import XML natively or shred the data into the system. These tools are intended for design of one database solution, thus generation and evaluation of alternative solutions becomes time consuming.

Other interesting work regarding design of hybrid storage is the work by Moro et al. [38]. They address the problem by a database design tool based on a conceptual design language and provide guidelines for when parts of the XML structure should be translated or not. In our case the problem is a bit different. The work has similar goals to HShreX but in our case we want to use the already existing XML schema as a starting point.

VII. Towards a future system

We currently use HShreX for creation of hybrid storage models that allows us to compare and evaluate different storage alternatives. Our experience so far is that the system allows fast creation of alternate storage models and that it is easy to create the models that we want to test. However, our experiments so far have highlighted extended functionalities that would be of interest for future versions of the system.

One such is enhanced annotation functionality, for instance to change data typing and add indexes to the created data model. For the moment the system contains a rudimentary implementation for data typing while indexes must be created by hand after loading the model into the database.

The bottleneck of the system is querying for the different data models. This is due to the complexity of the generated data model and the many alternatives provided by SQL/XML. We are investigating ways of automating this process as well, the idea is to use an automatic query translator that suggest a SQL/XML query based on a XQuery query where the user can reformulate the translated query if desired. Currently, we have a solution for using XPath query capabilities within HShreX. This would allow the user to issue XPath expressions inside HShreX that correspond to the original XML data. HShreX will then consult its internal shredding information and query the database for the right data.

Our long term goal is to get a better understanding of how to shred XML into good hybrid data models that is easy to work with and provide an efficient storage model. The final goal is to make HShreX smarter about its shredding rules, i.e., to make HShreX have a more dynamic set of rules and also enable the user to inform HShreX about usage scenarios which would influence these rules. To reach this goal we would like to develop a system which by analyzing data and XML structure could propose different hybrid data models for the user to choose from.

This would also involve combining the two systems i.e. to enable graph functionality directly within HShreX. This could be achieved either by specialized annotations for nodes and edges or possibly also in this case by automatic analysis of the XML data and queries to allow HShreX to automatically detect substructures that should be imported to the graph engine. This would yield a system where the user can choose to store parts of data as graphs, relational or native XML and take advantage of all the possibilities depending on his needs.

For the future it would be interesting to introduce more advanced graph functionality demanded for many biological applications. There is currently a lot of research in specialized and efficient graph management for biological pathways, such as aligning pathways [11] and identifying target molecules for creation of drugs [12]. To extend our solution with this functionality we need to extend the graph functionality provided by the graph package. The main contribution of this paper, i.e. how to integrate the functionality with XQuery, would, however, be unaffected.

VIII. Conclusion

In this paper we present two tools allowing easy and efficient access and analysis to the large amount of graph related XML data available within systems biology. The first tool is specialised on providing analysis for graph data. A graph model for handling directed and undirected labeled graphs was introduced. Access to the graph model is realized through the XQuery environment. The user can define vertices and edges, execute algorithms and access the graph data as XML for further processing. This results in an efficient framework for processing graph views of XML data with a prototype implementation in eXist and JUNG. The second tools support the user in exploring more efficient storage and querying for XML data. The tool enables hybrid XML storage by adding annotations to the XML schema. We evaluate the tools and show that they provide efficient processing. In the end of the paper we discuss our results and discuss steps towards a future tool that combine the features of the current tools.

Acknowledgment

We acknowledge the financial support of the Center for Industrial Information Technology, the Swedish Research Council and the German Academic Exchange Service. We thank Nahid Shahmehri, Andreas Heuer, Adelinde Uhrmacher and Dr. Holger Meyer for supporting the thesis work which provides the foundation for this paper. We are also grateful to Juliana Freire for input and discussions regarding the HShreX tool.

References

[1] L. Strömbäck and S. Schmidt, "An Extension of XQuery for Graph Analysis of Biological Pathways." in *The First International Conference on Advances in Databases, Knowledge, and Data Applications, DBKDA*, 2009.

[2] H. Hermjakob, L. Montecchi-Palazzi, G. Bader, J. Wojcik, L. Salwinski, A. Ceol, S. Moore, S. Orchard, U. Sarkans, C. von Mering, B. Roechert, S. Poux, E. Jung, H. Mersch, P. Kersey, M. Lappe, Y. Li, R. Zeng, D. Rana, M. Nikolski, H. Husi, C. Brun, K. Shanker, S. Grant, C. Sander, P. Boork, W. Zhu, P. Akhilesh, A. Brazma, B. Jacq, M. Vidal, D. Sherman, P. Legrain, G. Cesareni, I. Xenarios, D. Eisenberg, B. Steipe, C. Hogue, and R. Apweiler, "The HUPO PSI's Molecular Interaction format - a community standard for the representation of protein interaction data," *Nature Biotechnology*, vol. 22, no. 2, pp. 177–183, 2004.

[3] F. Collins, E. Green, A. Guttmacher, and M. Guyer, "A vision for the future of genomics research," *Nature*, vol. 422, pp. 835–847, April 2003. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=2003Natur.422..835C

[4] G. Bader, I. Donaldson, C. Wolting, B. Oulette *et al.*, "BIND - The Biomolecular Network Database," *Nucleic Acids Research*, vol. 29, no. 1, pp. 242–245, 2001.

[5] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler, "IntAct - an open source molecular interaction database," *Nucleic Acids Research*, vol. 32, pp. D452–D455, 2004.

[6] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. Gopinath, G. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein, "Reactome: a knowledgebase of biological pathways," *Nucleic Acids Research*, vol. 33, no. D428-D432, 2005.

[7] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori, "The KEGG resources for deciphering the genome," *Nucleic Acids Research*, vol. 32, pp. D277–D280, 2004.

[8] P. Karp, M. Arnaud, J. Collado-Vides, J. Ingraham, I. Paulsen, and M. J. Saier, "The E. coli EcoCyc Database: No Longer Just a Metabolic Pathway Database," *ASM News*, vol. 70, no. 1, pp. 25–30, 2004.

[9] M. Hucka, A. Finney, H. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray, A. Cornish-Bowden, A. Cuellar, S. Dronov, E. Gilles, M. Ginkel, V. Gor, I. Goryanin, W. Hedley, T. Hodgman, J.-H. Hofmeyr, P. Hunter, N. Juty, J. Kasberger, A. Kremling, U. Kummer, N. L. Novère, L. Loew, D. Lucio, P. Mendes, E. Minch, E. Mjolsness, Y. Nakayama, M. Nelson, P. Nielsen, T. Sakurada, J. Schaff, B. Shapiro, T. Shimizu, H. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics*, vol. 19, no. 4, pp. 524–531, 2003.

[10] G. D. Bader and M. P. Cary, *BioPAX - Biological Pathways Exchange Language Level 2, Version 1.0 Documentation*, BioPAX workgroup, December 2005.

[11] F. Ay, T. Kahveci, and V. Crecy-Lagard, "Consistent alignment of metabolic pathways without any abstraction modeling." in *International Conference on Computational Systems Biology (CSB)*, 2008.

[12] P. Sridhar, B. Song, T. Kahveci, and S. Ranka, "Mining methabolic networks for optimal drug targets." in *Pacific Symposium on Biocomputing (PSB)*, 2008, pp. 291–302.

[13] L. Strömbäck, D. Hall, and P. Lambrix, "A review of standards for data exchange within systems biology." *Proteomics*, vol. 7, no. 6, pp. 857–867, March 2007. [Online]. Available: http://dx.doi.org/10.1002/pmic.200600438

[14] G. Cochrane, P. Aldebert, N. Althorpe, M. Andersson, W. Baker, A. Baldwin, K. Bates, S. Bhattacharyya, P. Browne, A. van den Broek, M. Castro, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, C. Kanz, T. Kulikova, C. Lee, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, M. McHale, H. McWilliam, G. Mukherjee, F. Nardone, M. P. Pastor, S. Sobhany, P. Stoehr, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler, "EMBL Nucleotide

Sequence Database: developments in 2005." *Nucleic Acids Res*, vol. 34, no. Database issue, January 2006. [Online]. Available: http://dx.doi.org/10.1093/nar/gkj130

[15] A. Garny, D. Nickerson, J. Cooper, R. W. dos Santos, A. Miller, S. McKeever, P. Nielsen, and P. Hunter, "CellML and associated tools and techniques," *Philosophical Transactions of the Royal Society A*, vol. 366(1878), pp. 3017–3043, 2008.

[16] P. Murray-Rust and H. S. Rzepa, "Chemical Markup, XML, and the World Wide Web. 4. CML Schema," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 3, pp. 757–772, May 2003. [Online]. Available: http://dx.doi.org/10.1021/ci0256541

[17] The UniProt Consortium, "The Universal Protein Resource (UniProt)," *Nucl. Acids Res.*, p. gkm895, 2007. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/gkm895v1

[18] P. T. Spellman, M. Miller, J. Stewart, C. Troup, U. Sarkans, S. Chervitz, D. Bernhart, G. Sherlock, C. Ball, M. Lepage, M. Swiatek, W. L. Marks, J. Goncalves, S. Markel, D. Iordan, M. Shojatalab, A. Pizarro, J. White, R. Hubley, E. Deutsch, M. Senger, B. J. Aronow, A. Robinson, D. Bassett, C. J. Stoeckert, and A. Brazma, "Design and implementation of microarray gene expression markup language (mage-ml)." *Genome Biol*, vol. 3, no. 9, August 2002. [Online]. Available: http://dx.doi.org/10.1186/gb-2002-3-9-research0046

[19] P. G. A. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler, K. Cheung, C. E. Costello, H. Hermjakob, S. Huang, R. K. Julian, E. Kapp, M. E. Mccomb, S. G. Oliver, G. Omenn, N. W. Paton, R. Simpson, R. Smith, C. F. Taylor, W. Zhu, and R. Aebersold, "A common open representation of mass spectrometry data and its application to proteomics research," *Nature Biotechnology*, vol. 22, no. 11, pp. 1459–1466, November 2004. [Online]. Available: http://dx.doi.org/10.1038/nbt1031

[20] S. Orchard, C. F. Taylor, H. Hermjakob, Weimin-Zhu, R. K. Julian, and R. Apweiler, "Advances in the development of common interchange standards for proteomic data." *Proteomics*, vol. 4, no. 8, pp. 2363–2365, Aug 2004.

[21] R. Stanislaus, L. H. Jiang, M. Swartz, J. Arthur, and J. S. Almeida, "An XML standard for the dissemination of annotated 2D gel electrophoresis data complemented with mass spectrometry results." *BMC Bioinformatics*, vol. 5, no. 1, January 2004. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-5-9

[22] A. Keller, J. Eng, N. Zhang, X.-J. Li, and R. Aebersold, "A uniform proteomics MS/MS analysis platform utilizing open XML file formats," *Molecular Systems Biology*, vol. 1, no. 1, pp. msb4 100 024–E1–msb4 100 024–E8, August 2005. [Online]. Available: http://dx.doi.org/10.1038/msb4100024

[23] E. Deutsch, "mzML: A single, unifying data format for mass spectrometer output," *Proteomics*, vol. 8, no. 14, pp. 2776–2777, 2008. [Online]. Available: http://dx.doi.org/10.1002/pmic.200890049

[24] "mzML 1.1.0 Specification," 2009. [Online]. Available: http://www.psidev.info/index.php?q=node/257

[25] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron, "Minimum information about a microarray experiment (MIAME)-toward standards for microarray data." *Nat Genet*, vol. 29, no. 4, pp. 365–371, December 2001. [Online]. Available: http://dx.doi.org/10.1038/ng1201-365

[26] S. Orchard, H. Hermjakob, R. K. Julian, K. Runte, D. Sherman, J. Wojcik, W. Zhu, and R. Apweiler, "Common interchange standards for proteomics data: Public availability of tools and schema." *Proteomics*, vol. 4, no. 2, pp. 490–491, February 2004. [Online]. Available: http://dx.doi.org/10.1002/pmic.200300694

[27] N. Le Novère, A. Finney, M. Hucka, U. S. Bhalla, F. Campagne, J. Collado-Vides, E. J. Crampin, M. Halstead, E. Klipp, P. Mendes, P. Nielsen, H. Sauro, B. Shapiro, J. L. Snoep, H. D. Spence, and B. L. Wanner, "Minimum information requested in the annotation of biochemical models (miriam)," *Nature Biotechnology*, vol. 23, no. 12, pp. 1509–1515, December 2005. [Online]. Available: http://dx.doi.org/10.1038/nbt1156

[28] T. cytoscape consortium, 2009. [Online]. Available: www.cytoscape.org

[29] F. Bois, "GNU MCSim: Bayesian statistical inference for SBML-coded system biology models," *Bioinformatics*, vol. 25, no. 11, pp. 1453–1454, 2009.

[30] L. Strömbäck and D. Hall, "An Evaluation of the Use of XML for Representation, Querying, and Analysis of Molecular pathways." in *EDBT Workshops.*, 2006.

[31] L. Strömbäck and P. Lambrix, "Representation of molecular pathways: an evaluation of SBML, PSI MI and BioPAX," *Bioinformatics*, vol. 21, no. 24, pp. 4401–4407, October 2005.

[32] W3C, "XQuery 1.0: An XML Query Language." W3C, 2007. [Online]. Available: www.w3.org/TR/2007/REC-xquery-20070123/.

[33] K. Beyer, F. Özcan, S. Saiprasad *et al.*, "DB2/XML:Designing for Evolution." in *SIGMOD 2005*, 2005, pp. 31–38.

[34] M. Rys, "XML and relational Management Systems; Inside Microsoft SQL Server 2005." in *SIGMOD 2005*, 2005.

[35] R. Murthy, Z. Hua Liu, M. Krishnaprasad, S. Chandrasekar, A.-T. Tran, E. Sedlar, D. Flurescu, S. Kotsovos, N. Agarwal, V. Arora, and V. Krishnamurthy, "Towards an enterprise XML architecture," in *SIGMOD 2005*, 2005.

[36] M. Krishnaprasad, Z. Hua Liu, A. Manikutty, J. Warner, and V. Arora, "Native XQuery processing in Oracle XMLDB," in *SIGMOD 2005*, 2005.

[37] Z. Hua Liu, M. Krishnaprasad, and V. Arora, "Native XQuery Processing in XMLDB," in *SIGMOD 2005*, 2005.

[38] M. Moro, L. Lim, and Y.-C. Chang, "Schema Advisor for Hybrid Relational-XML DBMS." in *SIGMOD 2007*, 2007.

[39] S. Amer-Yahia, F. Du, and J. Freire, "A Comprehensive Solution to the XML-to-Relational Mapping Problem." in *ACM International Workshop on Web Information and Data Management (WIDM)*, 2004, pp. 31–38.

[40] D. Chamberlin, M. Carey, M. Fernandez, D. Florescu, G. Ghelli, D. Kossmann, J. Robie, and J. Simeon, "XQueryP: An XML Application Development Language." in *XML 2006*, 2006.

[41] D. Chamberlin, M. Carey, D. Florescu, D. Kossman, and J. Robie, "XQueryP: Programming with XQuery." in *Third International Workshop on XQuery Implementation, Experience, and Perspectives.*, 2006.

[42] G. Ghelli, C. R, and S. J., "XQuery!: An XML query language with side effects," in *Second International Workshop on Database Technologies for Handling XML Information on the Web (DataX 2006)*, 2006.

[43] L. Strömbäck, M. Åsberg, and D. Hall, "HShreX: a Tool for Design and Evaluation of Hybrid XML Storage," in *FLexDBIST 2009*, 2009.

[44] Mark Logic Corporation, "Mark Logic Server, XQuery API Documentation," Mark Logic Corporation, 2007. [Online]. Available: http://xqzone.marklogic.com/pubs/3.0/apidocs/Extension.html

[45] Oracle, "Oracle life science platform," Oracle, 2007. [Online]. Available: www.oracle.com/technology/industries/life_sciences/

[46] B. Eckman and P. Brown, "An overview of data models for the analysis of biochemical pathways." *Systems Biology*, vol. 50, no. 1, pp. 246–259, 2006.

[47] B. Bohannon, J. Freire, P. Roy *et al.*, "From XML Schema to Relations: A Cost-Based Approach to XML Storage." in *IEEE International Conference on Data Engineering*, 2002, pp. 64–75.

[48] D. Floresco and D. Kossman, "Storing and Querying XML data using RDBMS." *IEEE Data Eng. Bull.*, vol. 22, no. 3, pp. 27–34, 1999.

[49] H. Georgiadis and V. Vassalos, "XPath on stereoids: Exploiting relational engines for XPath performance." in *SIGMOD'07*, 2007.

[50] I. Mlynkova, "Standing on the Shoulders of Ants: Towards More Efficient XML-to-Relational Mapping Strategies." in *19th International Workshop on Databaseand Expert Systems Applications,*, 2009.

[51] T. Grust, J. Rittinger, and J. Teubner, "Why Off-the-shelf RDBMSs are better at XPath Than you Might Expect." in *SIGMOD'07*, 2007.

A Workflow System for Data Processing on Virtual Resources

Rainer Schmidt, Christian Sadilek, and Ross King
AIT Austrian Institute of Technology
Department of Safety & Security
Digital Memory Engineering
Donau-City-Str. 1, 1220 Vienna, Austria
firstname.lastname@ait.ac.at

***Abstract*—This paper describes challenges and approaches that have been addressed during the development of a workflow environment for digital preservation. The system addresses the general problem of efficiently processing collections of binary data using commodity software tools. We present a prototype implementation of a job execution service that is capable of providing access to clusters of virtual machines based on standard grid mechanisms. The service allows clients to specify individual tools and execute them in parallel on large volumes of data. This approach allows one to utilize a cloud infrastructure that is based on platform virtualization as a scaling environment for the execution of complex workflows. Here, we outline the architecture of the workflow environment, introduce its programming model, and describe the service enactment. With this paper we extend work previously presented in [1].**

***Keywords*-data intensive computing; cloud computing; service-oriented, workflow; digital preservation;**

I. Introduction

Due to rapid changes in information technology, a significant fraction of digital data, documents, and records are doomed to become uninterpretable bit-streams within short time periods. Digital Preservation deals with the long-term storage, access, and maintenance of digital data objects. In order to prevent a loss of information, digital libraries and archives are increasingly faced with the need to electronically preserve large volumes of data while having limited computational resources in-house. However, due to the potentially immense data sets and computationally intensive tasks involved, preservation systems have become a recognized challenge for e-science [2]. Preservation systems must be scalable in order to cope with enormous data volumes, for example such as are produced in fields like science and the humanities. Here, we argue that grid and cloud technology can provide the crucial technology for building scalable preservation systems.

The Planets project [1] aims to provide a service-based solution to ensure long-term access to the growing collections of digital cultural heritage data. The system supports the development, evaluation, and execution of preservation processes based on atomic software components. Components that perform preservation actions often rely on third-party tools (e.g. a file format converter) that must be pre-installed on a specific hosting platform. Planets provides an integrated environment for seamlessy accessing those tools based on defined service interfaces. The workflow execution engine implements the component-oriented enactor that governs life-cycle operation of the various preservation components, such as instantiation, communication, and data provenance. It allows the user to create distributed preservation workflows from high-level components that encapsulate the underlying protocol layers.

A crucial aspect of the preservation system is the establishment of a distributed, reliable, and scalable computational tier. A typical preservation workflow may consist of a set of components for data characterization, migration, and verification, and may be applied to millions of digital objects. In principle, these workflows could be easily parallelized and run in a massively parallel environment. However, the fact that preservation tools often rely on closed source, third-party libraries and applications that often require a platform-dependent and non-trivial installation procedure prevents the utilization of standard high performance computing (HPC) facilities. In order to efficiently execute a preservation plan, a varying set of preservation tools would need to be available on a scalable number of computational nodes. The solution proposed in this paper tackles this problem by incorporating hardware virtualization, allowing us to instantiate sets of transient system images on demand, which are federated as a virtualized cluster. The presented Job Submission Service (JSS) is utilized as the computational tier of a digital preservation system. Jobs are capable of executing data-intensive preservation workflows by utilizing a MapReduce [3] implementation that is instantiated within a utility cloud infrastructure. The presented system is based on the Planets Interoperability Framework, Apache Hadoop [4], and a JSS prototype providing a grid middleware layer on top of the AWS [2] cloud infrastructure.

In this paper, we present on an execution service for

[1] Preservation and Long-term Access through Networked Services, http://www.planets-project.eu/

[2] Amazon Web Services

preservation tools which relies on standard grid mechanisms and protocols like the Job Submission Description Language [5] (JSDL) and the HPC basic web service profile (HPCBP) [6]. We outline the architecture of the Planets workflow environment and introduce an XML-based workflow language that is designed to integrate complex service interaction based on reusable software components. Finally, we present experimental results that have been conducted using the Amazon Simple Storage Service (S3) and Elastic Compute Cloud (EC2) services (AWS) [7]. The paper is organized as follows: In section II we provide an overview of related work in the area of cloud and virtual computing, grids, and digital preservation, section III outlines the problem domain, section IV presents the architecture of the workflow environment, in section V, we introduce the workflow model and language, section VI presents the Job Submission Service and its prototype implementation, section VII reports experimental results, and section VIII concludes the paper.

II. Background and Related Work

A. Cloud and Virtual Computing

The demand for storage and computational power of scientific computations often exceeds the resources that are available locally. Grid infrastructures, services and remote HPC facilities can provide a viable solution for scientists to overcome these limitations. However, many applications require dedicated platforms or need time-consuming adaptations in order to utilize a remote resource. Virtual machine technology provides software that virtualizes a physical host machine, allowing the deployment of platform-independent system images. The deployment of virtual computer instances is supported by a virtual machine monitor, also called a hypervisor. Cloud systems are consumable via Internet-based services offering IT-technology in the form of applications, hosting platforms, or access to computer infrastructures. Amazon's EC2 and S3 services, one of the most prominent commercial offerings, allow users to rent large computational and storage resources on-demand. EC2 is based on the Xen [8] [9] hypervisor allowing one to prepare and deploy virtual system instances that suit individual application needs. S3 provides access to a global, distributed, and replicated storage system. A detailed evaluation of Amazon's compute, storage, and coordination (SQS) web services and their suitability for scientific computing is given in [10] [11]. Deelman et al. provides cost-based analysis of utilizing the Amazon cloud infrastructure for scientific computing [12]. A proof-of-concept study that runs a complex nuclear physics application on a set of virtual machine nodes is presented in [13]. The Nimbus workspace cloud provides a service to scientific communities allowing the provisioning of customized compute nodes in the form of Xen virtual machines that are deployed on physical nodes of a cluster [14]. A study that compares differences of grid and cloud systems that is based on Amazon's EC2 and S3 services is given in [15]. An experiment were a large set of scanned newspaper articles haven been converted to PDF documents using the Amazon cloud infrastructure has been reported in [16].

B. Distributed Data Infrastructures

Research in fields like high-energy physics and earth science produce large amounts of irreplaceable data that must be accessed and preserved over time. For example, in earth observation, data is typically geographically dispersed over different archive and acquisition sites, using a multitude of data and meta-data formats [17]. Grid systems provide dependable access and the coordinated resource sharing across different organizational domains [18]. Data grids [19] focus on the controlled sharing and management of large data sets that are distributed over heterogeneous sites and organizations. In this context, an important aspect is the storage of data in a reliable, distributed, and replicated way. Preservation archives are systems that aim to implement long-term preservation in order to manage data integrity and technological evolution. This includes migrating digital objects to new technologies, maintaining their relationships and preservation metadata. Data grids can be used as the underlying technology to implement digital libraries and distributed preservation archives [20]. The Storage Resource Broker (SRB) [21] of the San Diego Supercomputer center implements a distributed data management environment for data collections based on a virtual file system, logical namespaces, and a metadata repository (MCAT). The iRODS system extends SRB by an adaptive rule system to enforce data management policies based on server-sided micro services [22]. The Transcontinental Persistent Archives Prototype (TPAP) [23] provides a testbed across a number of independent US sites that are linked by high-performance network (DREN), allowing the distribution of electronic records across multiple institutions based on SDSC's SRB. An effort to develop a service-oriented infrastructure for the automated processing of linguistic resources effort is undertaken by the Clarin project [3]. Computational grid systems provide a complimentary technology and are often combined with data grids. For example, the EGEE project [24], currently the world's largest production grid, provides large quantities of distributed CPUs and petabytes of storage. A survey of initiatives that focus on the integration of emerging technologies like digital libraries, grid, and web services for distributed processing and long-term preservation of scientific knowledge is given in [25].

III. Overview

The Planets infrastructure aims to provide an e-research and problem-solving environment for the development

[3] www.clarin.eu

of preservation workflows that supports flexible tool and workflow integration. It supports the planning as well as the execution and evaluation of repeatable preservation experiments. This preservation environment is implemented as a service-oriented architecture that is accessible by users via a portal server. The graphical end-user applications typically implement a scientific experimentation process and access the workflow execution engine (WEE) as part of the portal environment. A major challenge of the workflow execution engine is the enactment of a broad range of experiments that tremendously vary in complexity and scale. Experiments may be performed based on local desktop components, remote application services, as well as by incorporating large-scale compute and storage resources. The workflow environment and execution service presented in this paper addresses the following research issues:

- A grid service that provides access to a variety of third-party tools based on clusters of customized virtual images.
- The incorporation of data intensive computation mechanisms for the efficient processing of non-textual artifacts.
- A high-level workflow language for the task-parallel execution of (parallel) compute jobs on different middleware systems.

IV. The Workflow Environment

This section outlines the workflow execution engine, it's service interaction mechanisms, as well as the programming interface.

A. The Workflow Execution Engine

In the following, we outline the basic interaction pattern between the user application, the workflow environment, and the Job Execution Service. A detailed discussion of the Planets workflow system and its implementation is beyond the scope of this paper. The sequence diagram in Fig. 1 schematically depicts the interaction of a workflow client (*Preservation Application*), the workflow service API (*Workflow Execution Engine*), and the generic service proxy (*Execution Manager*) during workflow execution. The workflow service basically provides SOAP interfaces for the submission and monitoring of workflow processes. A workflow document provides an XML-based description of an executable process (section V), which is typically generated by a workflow editor and/or a domain specific graphical application that utilizes the workflow service. The workflow designer (application) is expected to lookup and select the required services, tools, and job parameters based on the Planets service and tool registries, which provide graphical as well as SOAP interfaces. In its current implementation, the workflow execution engine does not provide advanced resource management capabilities like on-demand service selection, dynamic resource allocations, or quality of service support. After a client has submitted a workflow description for execution, an identifier is returned and the control is handed over to the workflow execution engine. The WEE enqueues the workflow and starts the execution once all required preconditions are met. Resources are limited to the number of overall available cloud nodes and a maximum number of concurrently running workflows. A workflow preprocessing stage (*prepare Workflow*) validates the workflow document and evaluates the resource demand. During workflow execution, each activity is associated with an *Execution Context*, which provides a space that links an ongoing activity (and all its metadata) with the corresponding workflow instance. This includes information such as the service interface, endpoint, tool configuration, walltime, as well as a pointer to the result object. The implementation of the *Execution Context* is specific to the the *Execution Service* that is invoked. At this stage of development, three types of execution services are supported (see Fig. 2). The *LocalExecutionManager* executes local Java components which are typically used for implementing metadata operations and decision logic. The *WebServiceExecutionManager* is used to dynamically invoke remote preservation services. These services implement a predefined Web service profile, which is invoked by utilizing the Web Services Interoperability Technology (WSIT) [4] framework. Planets preservation services implement interfaces and messaging protocols for operations such as file characterization, modification, migration, validation, or comparison [26]. The *EC2ExecutionManager* implements the invocation and message exchange with the job submission service. This service implements a grid service profile and is used to execute long-running and data intensive jobs (section VII). Furthermore, the workflow execution engine provides a method for status inquiry and may send an email notification upon the completion of a workflow.

B. Programming Interface

Planets preservation workflows are build from Java components, allowing a workflow developer to assemble typical preservation cases from atomic services. The workflow API defines a set of functional interfaces that allow users to easily assemble and executable preservation workflows including preservation services like *migrate*, *characterize*, *compare*, or *validate*. The interfaces are compatible among each other and operate based on a minimal data abstraction, called a *digital objects*. Hence, on the API level each service consumes and produces a digital object. A digital object holds metadata like technical, provenance, or other preserva-

[4] https://wsit.dev.java.net/

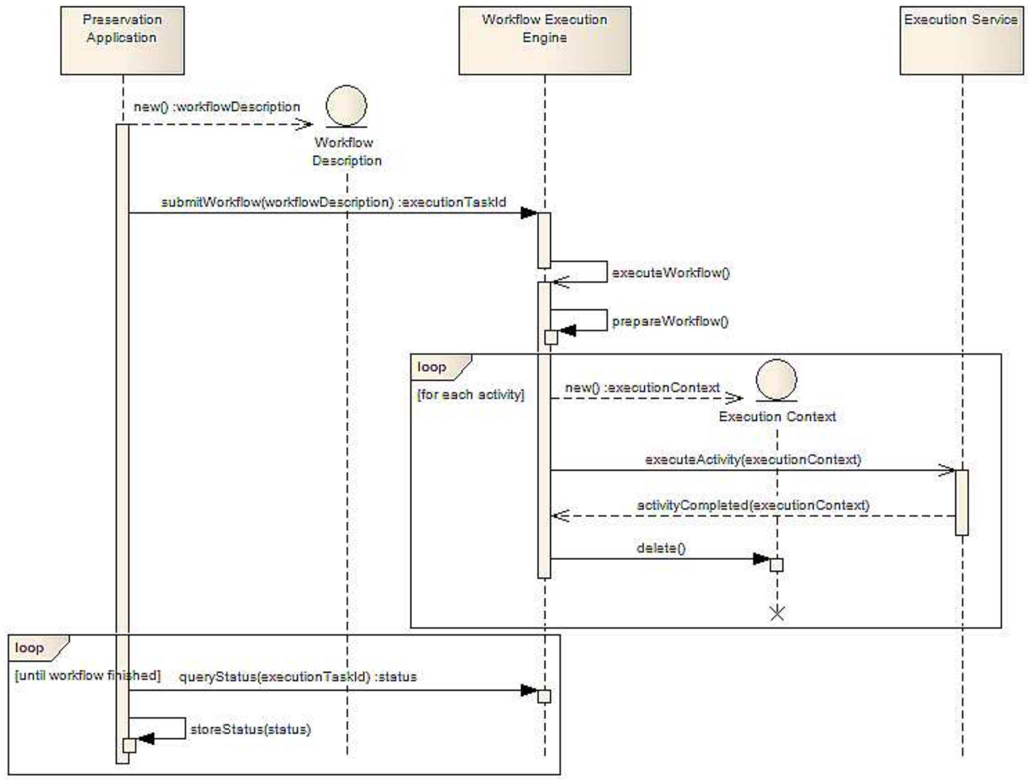


Figure 1. Sequence diagram showing the interactions between a Preservation Application, the Workflow Execution Engine, and the Execution Manager during workflow execution.

tion information about a digital resource including a handle to the actual data. Digital objects can be passed between different preservation services and point to different types of digital resources (e.g. files, collections, archives). The preservation metadata of processed digital objects must be be handled on the workflow level and is managed by trusted Java components.

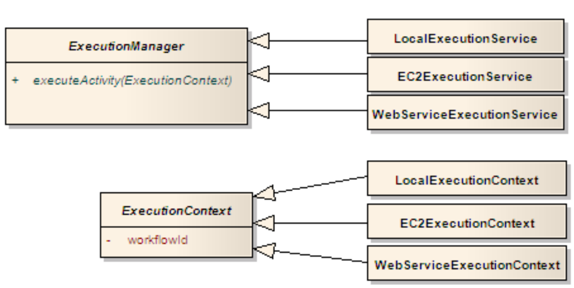


Figure 2. Class diagram showing different implementations of the abstract Execution Manager and Execution Context classes.

V. The Workflow Model

A. Objectives

In this section, we present a resource intensive preservation workflow that can be executed by employing the Planets Job submission service. Such a workflow requires a complex control logic, which must be defined and executed by the workflow system. In section IV-B, we outline a workflow API that abstracts away low-level details such as service interfaces and messaging protocols from the workflow developer. These components could be easily assembled into executable workflow based on the natural programming language (i.e. Java). However, for reasons like simplicity, robustness (e.g. checkpointing and restart), and platform independence, workflows should be defined in a declarative fashion. In section V-C, we introduce initial developments on an XML-based workflow language for orchestrating Planets preservation services, in particular the JSS. Work on this workflow environment is influenced by a number of existing web/grid service workflow systems including DAGMan [27], Triana [28], and GridAnt [29].

B. Use Case and Data Flow

The typical preservation use-case we are targeting is the processing of large data collections. A collection describes

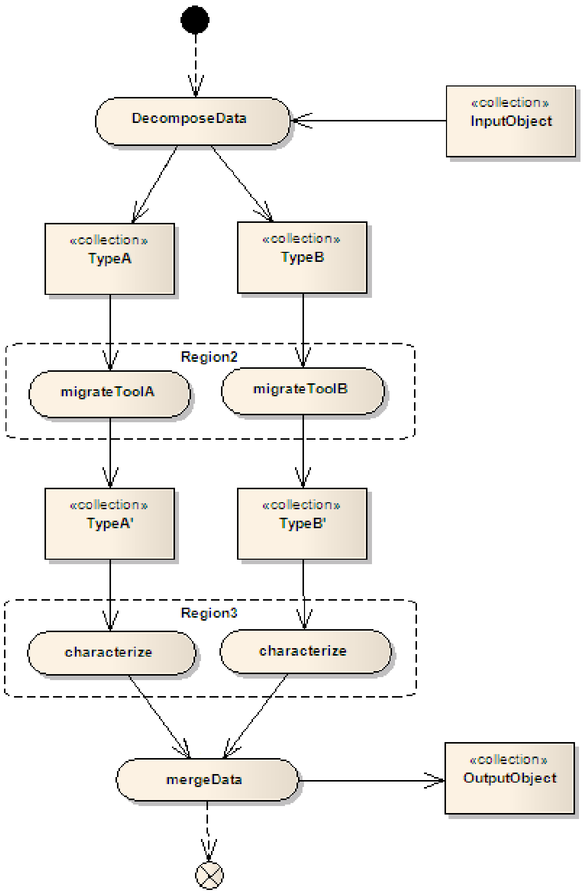


Figure 3. Data flow for a simple bulk processing application: data objects are physical and referenceable entities in a data store, activities are executed on parallel hardware, regions need to be co-scheduled. Messaging, metadata management, control flow and decision logic are not shown in the diagram.

data that is logically interrelated and described using some metadata language. If a collection is organized and curated within a digital repository system, it must be exported first before it can be processed by the presented preservation system. A major difficulty for data preservation in general arises from the diversity of digital data resources and methods to store, describe, and organize them. Examples of data collections we aim to preserve range from simple file collections, over data organized using some markup language (XML, HTML), to data organized in triple stores (RDF/XML).

Figure 3 shows the data flow graph for a simple preservation use case. Parallel *Regions* indicate that one or many tasks might be executed as data parallel jobs. Consider a collection of scanned book pages and associated text documents for a historic book collection, organized by a set of XML files. The idea of the preservation workflow is to convert all images into the JPEG 2000 format and all documents into the PDF/A format. The process flow works as follows; first separate the data into images and documents based on filename extensions. This can be done by running an application (script) for each desired output type. The activity takes a handle to the input data (for example pointing to an S3 bucket) as input and produces a data handle for each output type, represented as collection A and B in the diagram. Once the data is sorted, a migration task is started for each file set using a tool like ImageMagick [5] for the image migration and another tool for PDF/A document conversion. Both activities should be run as parallel jobs in order to minimize execution time. Therefore, each of the migration tasks is launched as a parallel job that executes on a specified number of (e.g. hadoop) cluster nodes. After both migration tasks are finished, collection A' and B' are created. In the next step, one needs to verify the format of the resulting files and extracts relevant properties like file size, image size, or the number of pages. This is done by starting two parallel jobs that invoke a characterization tool like jHove [6] using a handle to collection A' and B' as input. In the final step the data collections are merged and an updated version of the XML records linking to the new data manifestations are generated.

It is important to note that the dataflow graph does not represent the workflow programming model. The presented workflow execution engine follows a more service-oriented approach where the execution services are orchestrated by the WEE during execution time. Hence, a continuous message exchange between workflow execution engine and the preservation services is required. Such a model gives the workflow execution engine much more control over the execution during runtime as compared to batch submission of workflow graphs. This adds additional communication overhead to the overall system but allows one to implement much more complex workflow logic. This is for example required in order to implement decision logic that depends on metadata that is generated and evaluated during runtime.

C. Control Flow

Although a final data flow - as shown in figure 3 - results in a Directed Acyclic Graph (DAG), many workflows cannot be specified in this way. In order to define such processes, it is important to be able to express control logic like conditions or iterations. For a typical preservation workflow that is executed within this environment, it is for example required to evaluate intermediate results or implement error handling. In the following, we describe first results in defining an XML-based workflow language for data-intensive preservation workflows. These workflows can include activities that are local, distributed and/or executed on parallel hardware (i.e. through the JSS). A major design goal is to foster simplicity of the language based on reusable

[5] http://www.imagemagick.org/
[6] http://hul.harvard.edu/jhove/

software components. Therefore, our approach is to encapsulate the complexity of interacting with the system within an extensible set of high-level Java components. A workflow can be assembled by interlinking these component based on an XML document. We employ two abstraction layers: (a) reusable Java components for implementing complex logical tasks and (b) an XML schema for interlinking these components. This approach can be contrasted to the approach taken by low-level service orchestration languages like WS-BPEL. WS-BPEL [7] provides a very precise language that allows the specification of web service interactions at a messaging level using Web service standard languages like BPEL, XML, XPath. However, creating BPEL-based workflow documents can become a difficult and error prone task which is difficult to automate. The presented approach is less universal but designed with the idea in mind to be easily supported by a graphical editor.

D. Example

Figure 4 provides an example workflow snippet for the exection of two activities using the Job Submission Service (JSS). Both services are concurrently executed using the *execute* command. The command does not block the program execution until a corresponding *receive* operation is issued (similar to MPI [8] send/receive). The service is specified by its endpoint address as well as a proxy component (*class*) that implements the interaction with a certain service interface. Furthermore, the preservation service needs to be configured by a list of name-value pairs. The required parameters depend on the service implementation (published within the service registry), which specify the underlying application/tool, specific arguments, or the resource demands (e.g. number of nodes). In case of the execution service this information is required to automatically generate the job descriptor. The service execution is furthermore associated with a handle (*puid*) to the *digital object* representation of the input data. Digital objects contain provenance and other metadata about a physical data entity and are organized within a metadata repository. The *receive* operation blocks the workflow until the corresponding service execution has been completed and a resulting *digital object* has been created. The object represents the result of a preservation service, which might be enriched metadata (e.g. by a characterization) or the generation of new data items (e.g. migration, modification). Methods for evaluation and storing digital objects are implemented by the metadata repository API.

E. File Transfer

A significant research challenge in executing Grid workflows is the transfer of large files between activities. This is in particular true, when the data needs to be transferred between different sites during workflow execution. For the presented experiments, we exploit a utility cloud for running data-intensive experiments and only transfer metadata during workflow execution. The data resides within an virtual storage environment (S3) and is processed by a range of parallel applications.

```xml
<!-- executing two remote jobs -->
<!-- parallel -->
  <execute id="1">
    <service>
      <class>planets.JSS</class>
      <endpoint>http://...</endpoint>
      <parameters>
        <name>app</name>
        <value>cmdexec.jar</value>
        <name>tool</name>
        <value>/bin/tools/econv.sh</value>
        <name>data_in</name>
        <value>http://s3...</value>
        <name>data_out</name>
        <value>http://s3...</value>
        <name>nnodes</name>
        <value>4</value>
      </parameters>
    </service>
    <puid>puid://s3data/...</puid>
  </execute>
  <execute id="2">
    <!-- declare job2 here-->
  </execute>
  <receive id="1">
    <digObj>digObjA</digObj>
  </receive>
  <receive id="2">
    <digObj>digObjB</digObj>
  </receive>
<!-- /parallel -->
<!-- validating result digObjs. -->
```

Figure 4. XML workflow declaration for execution two concurrently running services. The workflow execution is blocked until both services complete by corresponding receive operations.

VI. The Job Submission Service

A. Motivation

In the context of grid computing and data grids, digital preservation archives are systems that can preserve the output of computational grid processes [20]. An important issue in the context of preserving existing digital content is the process of deriving metadata from digital assets like file collections in order to extract significant semantic information for their preservation (e.g. format characterization). Decisions in preservation planning [30] rely on information that needs to be generated by algorithms and tools for feature extraction, format identification, characterization, and validation [31]. Migrating digital entities between different

[7] www.oasis-open.org/committees/wsbpel/
[8] http://www.mpi-forum.org/docs/

formats typically relies on sequential, third-party libraries and tools that are not supported by scientific parallel and grid systems. Therefore, we propose a service that employs clusters of customizable virtual nodes in order to overcome these restrictions. The IF JSS implements a grid service that provides access to a virtual cluster of large numbers of individually tailored compute nodes that can process bulk data based on data-intensive computing mechanisms and that is integratable with computational and data grid systems.

B. Web Service Profile

Developing an infrastructure for digital preservation involves many grid-specific aspects including the processing of large volumes of data, conducting experiments in distributed and heterogeneous environments, and executing workflows that cross administrative and institutional boundaries. The service presented in this paper focuses on the aspect of submitting and executing data-intensive jobs as part of a digital preservation infrastructure. In order to be able to take advantage of existing grid solutions and to promote interoperability and integration, the IF JSS service is based on a standard grid service profile (HPCBP) for job scheduling (called the basic HPC use case) that is being well adopted by scientific and industrial systems [32]. The OGF Basic Execution Service (BES) [33] defines Web service interfaces for starting, managing, and stopping computational processes. Clients define computational activities in a grid based on JSDL documents. The OGF HPC Basic Profile (HPCBP) specification defines how to submit, monitor, and manage jobs using standard mechanisms that are compliant across different job schedulers and grid middlewares by leveraging standards like BES, JSDL, and SOAP. Our current implementation provides interfaces that support the BES base case specification and accept JSDL documents that are compliant with the HPCBP profile.

C. Basic Service Components

The Job Submission Service (JSS) prototype has been implemented based on a set of exchangeable core components, which are described below. The JSS is a stand-alone Web Service deployed in a Java EE Web Container as shown in Fig. 5. It is secured using HTTPS and SSL/TLS for the transport-layer and WS-Security based on X.509 server certificates and username/password client credentials for the message-layer. In order to submit a request to the JSS, username and password have to be provided that match a previously created account for the institution that utilizes the service. The individual accounts, utilization history, and potentially billing information are maintained by the *Account Manager* component. As HPCBP is used as the web service profile, JSDL documents are used to describe the individual job requests which need to be mapped to physical resources by the resource manager. The *JSDL parser* component validates the XML document and creates an object structure that serves as input for the *Execution Manager*. A *Session Handler* maps service requests based on activity identifiers to physical jobs and keeps track of their current status (e.g. pending, running, finished, failed). The *Execution Manager* interfaces with three components *the Handle Resolver*, *Input Generator*, and *Job Manager* that depend on the resource manager implementation, which is provided by Apache Hadoop in our case. The file handle resolver is used to validate a logical file handle (a URI) and resolve the physical and accessible data reference. The next step is the generation of an input file for a bulk of data that needs to be processed by a parallel application utilizing a particular preservation tool. Finally, the Job Manager prepares a job script and schedules a job using the resource manager.

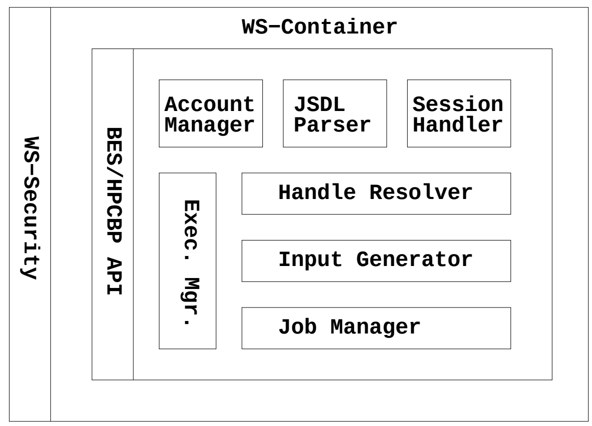


Figure 5. Job Submission Service Components

D. Implementation for MapReduce and Amazon's EC2, and S3 Services

The experimental results presented in section VII have been conducted using an *Execution Manager* implementation for (1) the Hadoop resource manager, (2) Amazon's EC2 compute cloud, and (3) the S3 storage infrastructure. In principle, each of the aforementioned components could be exchanged by different implementations and be connected to different resources, for example a local (e.g Condor [34] based) workstation cluster and network file system. In the following, we describe the functionality of the "cloud-enabled" execution manager. A file handle resolver is used to map a logical handle of a data collection to physical references that are meaningful for the application that needs to access the data (e.g. a file URI, a HTTP URL). Our file handle resolver is implemented in a way that it utilizes the S3 REST-based API to simply generate a list of URIs for files that are contained within an input bucket. The *Input Generator* uses this information to create an input file for the MapReduce application that processes the input data. *MapReduce* is a framework and programming model that has been introduced by Google to support parallel data-intensive computations. Apache Hadoop is an open source MapReduce implementation that can be used to cluster commodity computers. Also, Hadoop provides built-in support for EC2 and S3. We use Hadoop's own distributed file system to store input files across the computing nodes. The *Job Manager* component passes the input file together with an MapReduce application (the *CommandExecuter*) and information extracted from the JSDL object to the Hadoop job scheduler. The *CommandExecuter* is responsible for handling the S3 bulk data i/o, processing the *input splits* based on pre-installed applications as specified by the user, and for output generation. Finally, the outputs produced by each node are merged to form the output data collection.

VII. Experimental Results

A. Preliminary Considerations

The experiments were carried out as a quantitative evaluation of utilizing a virtual, cloud-based infrastructures for executing digital preservation tools. For all experiments, a simple workflow was implemented that migrates one file collection into a new collection of a different format using the `ps2pdf` command-line tool. It is important to note that the selected tool is replaceable and not relevant for the presented experiments. Four dimensions have been analyzed and compared to sequential executions on local execution environments: the execution time, the number of tasks, the number of computing nodes, the physical size of the digital collections to migrate. As performance metrics we calculate Speedup and Efficiency [35] as formally described in equations $S_{s,n}$ (1) and E_p (2).

$$S_{s,n} = Tseq_{s,n}/Tp_{s,n} \tag{1}$$

$$E_p = S_{s,n}/p \tag{2}$$

where:
s - is the physical object size,
n - is the number of tasks,
p - is the number of computing nodes.
$Tseq$ - is the sequential execution time,
Tp - is the execution time with p computing nodes.

B. Experiment Setup

For the experiments, we utilized the Amazon Elastic Compute Cloud (EC2) as a cloud infrastructure, leasing up to 150 cluster nodes, each running a custom virtual images based on RedHat Fedora 8 i386, Apache Hadoop 0.18.0, and a set of pre-installed the migration tools. The used default system instances provide one virtual core with one *EC2 Compute Unit*, which is equivalent to the capacity of a 1.0-1.2 GHz 2007 Opteron or a 2007 Xeon processor. Bulk data was stored outside the compute nodes using Amazon's Simple Storage System (S3) due to scale and persistence considerations. We experienced an average download speed from S3 to EC2 of 32.5 MByte/s and an average upload speed from EC2 to S3 of 13.8 MByte/s at the Java level. At the time conducting the presented experiments, the per hour price for an EC2 default instance was $0.10.

C. Measurements and Results

For the experiments shown in Fig. 6 we executed all computations on a constant number of five virtual nodes. The number of migration tasks was increased using different sized digital collections to compare the execution time within EC2 to a sequential local execution (SLE) on a single node with identical hardware characteristics. Fig. 6 focuses on the intersection points of the corresponding curves for SLE and EC2 identifying the critical job size for which the parallel execution within EC2 is faster than the sequential execution on a local machine. The results including Speedup and Efficiency for jobs with a large task sizes outside the bounding box of Fig. 6 are shown in table I. For the experiments shown in Fig. 7 we held the number of tasks constant (migration of a set of one thousand 70kB files) and increased the number of computing nodes form 1 to 150 to evaluate scalability. The values for Speedup, Efficiency and execution time were calculated based on the sequential local execution time for a given parallel job. As shown in table II, Speedup increases significantly with an increasing number of nodes due to relatively small overheads of the data parallel application model (see VII-D).

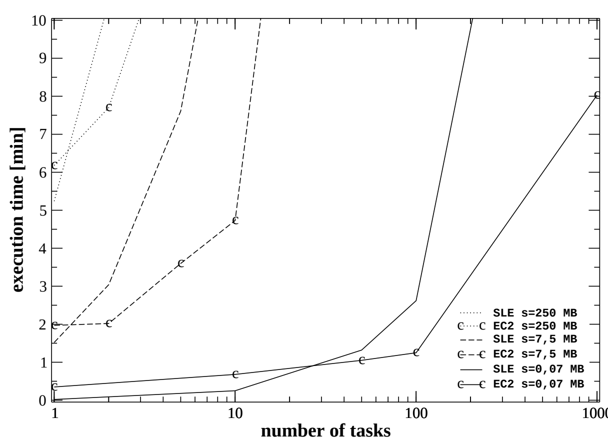


Figure 6. Execution time for an increasing number of migrations tasks and a constant number of computing nodes. The execution on five (EC2) nodes is compared to a sequential local execution (SLE) of the same task.

D. Interpretation of Results

Already for a small number of migration tasks the parallel execution within EC2 proved to be faster than the sequential execution on a single node (see Fig. 6). A Speedup of 4.4 was achieved for 5 nodes with n=1000 and s=7.5 MB (see table I) proving the suitability and potential of employing

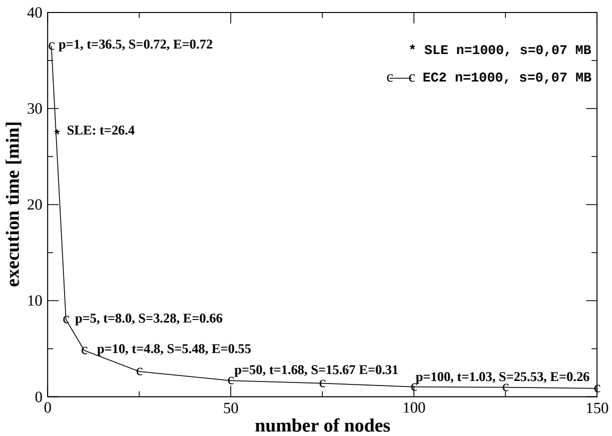


Figure 7. Execution time for 1000 constant migration tasks using an increasing number of computing nodes.

Tasks (n)	Size (s) [MB]	SLE exec. time [min]	EC2 exec. time [min]	$S_{s,n}$	E_p
1000	0.07	26.38	8.03	3.28	0.67
100	7.5	152.17	42.27	3.60	0.72
1000	7.5	1521.67	342.70	4.44	0.88
100	250	523.83	156.27	3.36	0.67
1000	250	5326.63	1572.73	3.37	0.68

Table I
RESULTS OUTSIDE THE BOUNDING BOX OF FIG. 6 INCLUDING SPEEDUP AND EFFICIENCY

(even small) clusters of virtual nodes for digital preservation of large data amounts. Results in Fig. II show that the system achieves good scalability when significantly increasing the number of utilized cluster nodes. However, following overheads which affect the efficiency of the described experiments have been identified: (1) Local execution (SLE) vs. cloud-based execution (p=1, n=1000). The master server for the Hadoop distributed file system which is running on a single worker node added 30% (8min) overhead on that node compared to an SLE (26min). We experienced less than

Number of nodes (p)	EC2 exec. time [min]	$S_{s,n}$	E_p
1	36.53	0.72	0.72
5	8.03	3.28	0.66
10	4.82	5.48	0.55
25	2.63	10.02	0.40
50	1.68	15.67	0.31
75	1.40	18.84	0.25
100	1.03	25.53	0.26
125	0.98	26.83	0.21
150	0.87	30.44	0.20

Table II
RESULTS SHOWN IN FIG. 7 COMPARED TO THE SEQUENTIAL LOCAL EXECUTION OF A GIVEN JOB (N=1000, S=0.07 MB) OF 26.38 MIN.

10% overhead introduced by S3 (compared to a local file system). (2) For a larger number of nodes (p $>$ 50, n=1000) efficiency decreases for various reasons, e.g. coordination. As all nodes are considered blocked until a job is processed, a large fraction of nodes are idle until the last process has finished. Also for short execution times per node, relatively small overheads like network delays and startup time have considerable impact on efficiency.

VIII. CONCLUSIONS AND FUTURE WORK

The emergence of utility cloud services introduced a novel paradigm for the provisioning of large-scale compute and storage resources [36]. Clouds allow their users to lease and utilize hard and software resources residing in large global data centers on-demand. This provides a generic model that can be exploited for business as well as for scientific applications. In the context of high-performance computing, it is obvious that such a model cannot replace dedicated clusters or other high-end and supercomputing facilities. However, it has been shown that applications in the area of data-intensive and high-throughput computing can be well applied to the cloud computing model [37]. Cloud infrastructures provide in general much less specific services than dedicated systems like compute clusters or Grid resources. The AWS EC2 service for example allows the user to control the software that is installed on the utilized virtual machines, commission and decommission computational resources on demand, and it does not require the user to wait for free instances/nodes before using them.

The integration of such resources into an infrastructure for distributed computing provides an important challenge in this context. It is important to identify the differences in orchestrating clouds compared to existing service-computing models. In this paper, we have presented a grid execution service that provides parallel processing of bulk data based on customizable virtual nodes as part of a digital preservation infrastructure. This service has been deployed and evaluated using Amazon's utility cloud infrastructure. We argue that building such computational services based on virtual images can provide a viable technology for the provisioning of domain-specific applications on a larger scale. Furthermore, we introduce work on a workflow system for the concurrent orchestration of cloud-based execution services. Future work will deal with the employment of a common authorization mechanism and protocol for secure web-based data access. In the area of digital libraries and archives, we feel that in particular, legal concerns, security policies, and SLAs will require extensive consideration. Another research goal will be the elaboration of resource management issues for on-demand computing. In particular, we will investigate in scheduling algorithms for distributing tasks across cloud nodes and clusters.

Acknowledgments

Work presented in this paper is partially supported by European Community under the Information Society Technologies (IST) 6th Framework Programme for RTD - Project Planets (IST-033789).

References

[1] R. Schmidt, C. Sadilek, R. King, "A Service for Data-Intensive Computations on Virtual Clusters," *First International Conference on Intensive Applications and Services (INTENSIVE09)*, 2009.

[2] Digital Preservation Coalition, "Digital Curation: digital archives, libraries, and e-science," Seminar report, London, UK, 19 October 2001, http://www.dpconline.org/graphics/events/ digitalarchives.html.

[3] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," in *Proceedings of OSDI'04*, 2004.

[4] Apache Hadoop, http://hadoop.apache.org/.

[5] A. Savva et. al, "Job Submission Description Language (JSDL) Specification, Version 1.0," Technical Report, 2005.

[6] B. Dillaway, M. Humphrey, C. Smith, M. Theimer, and G. Wasson, "HPC Basic Profile, v. 1.0. GFD-R-P.114," Technical Report, 2007.

[7] Amazon Web Services, http://aws.amazon.com.

[8] The Xen Project, http://xen.org/.

[9] Amazon Web Services, "Overview of security processes," June 2009, http://awsmedia.s3.amazonaws.com/pdf/ AWS_Security_Whitepaper.pdf.

[10] S. Garfinkel, "An Evaluation of Amazon's Grid Computing Services: EC2, S3 and SQS," Technical Report TR-08-07, Tech. Rep., 2007.

[11] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for science grids: a viable solution?" in *DADC '08*, 2008, pp. 55–64.

[12] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the Montage example," in *In Proceedings of SC'08*, 2008, pp. 1–12.

[13] K. Keahey, T. Freeman, J. Lauret, and D. Olson, "Virtual Workspaces for Scientific Applications," in *SciDAC 2007 Conference*, June 2007.

[14] Nimbus Cloud, http://workspace.globus.org/ clouds/nimbus.html.

[15] "An EGEE Comparative Study: Grids and Clouds Evolution or Revolution," 2008, https://edms.cern.ch/file/925013/3/EGEE-Grid-Cloud.pdf.

[16] D. Gottfrid, http://open.blogs.nytimes.com/ 2007/11/01.

[17] J. V. Bemmelen, L. Fusco, and V. Guidetti, "Access to Distributed Earth Science Data Supported by Emerging Technologies," in *EnviroInfo 2005*, September 2005.

[18] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organizations," *International Journal of Supercomputer Applications*, vol. 15, no. 3, 2001.

[19] S. Venugopal, R. Buyya, and K. Ramamohanarao, "A taxonomy of Data Grids for distributed data sharing, management, and processing," *ACM Comput. Surv.*, vol. 38, no. 1, p. 3, 2006.

[20] R. Moore, A. Rajasekar, and M. Wan, "Data Grids, Digital Libraries, and Persistent Archives: An Integrated Approach to Sharing, Publishing, and Archiving Data," *Proceedings of the IEEE*, vol. 93, no. 3, pp. 578–588, March 2005.

[21] A. Rajasekar, M. Wan, and R. Moore, "MySRB & SRB: Components of a Data Grid," in *HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, 2002, p. 301.

[22] M. Hedges, A. Hasan, and T. Blanke, "Curation and Preservation of Research Data in an iRODS Data Grid," in *E-SCIENCE '07: Proceedings of the Third IEEE International Conference on e-Science and Grid Computing*, 2007, pp. 457–464.

[23] The U.S. National Archives and Records Administration, "The Transcontinental Persistent Archives Prototype (TPAP)." [Online]. Available: http://www.archives.gov/era/research/tpap.html

[24] Enabling Grids for E-sciencE (EGEE), http://project.eu-egee.org.

[25] L. Fusco, J. van Bemmelen, and V. Guidetti, "Emerging technologies in support of long-term data and knowledge preservation for the earth science community," in *PV 2005*.

[26] R. Schmidt, R. King, A. Jackson, C. Wilson, F. Steeg, and P. Melms, "A framework for distributed preservation workflows," in *Proceedings of The Sixth International Conference on Preservation of Digital Objects (iPRES)*, San Francisco, USA, 2009.

[27] P. Couvares, T. Kosar, A. Roy, J. Weber, and K. Wenger, "Workflow Management in Condor," in *Workflows for e-Science*, 2007, pp. 357–375.

[28] I. Taylor, I. Wang, M. Shields, and S. Majithia, "Distributed computing with Triana on the Grid," *Concurrency and Computation:Practice and Experience*, vol. 17, no. 1–18, 2005.

[29] K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi, "Gridant: a client-controllable grid workflow system," 2004, pp. 10 pp.+.

[30] *Reference Model for an Open Archival Information System (OAIS), Blue Book, Issue 1*, CCSDS - Consultative Committee for Space Data Systems, January 2002.

[31] C. Chou, "Format Identification, Validation, Characterization and Transformation in DAITSS," in *Proceedings of IS&T Archiving 2007*, May 2007, pp. 151–156.

[32] C. Smith, T. Kielmann, S. Newhouse, and M. Humphrey, “The hpc basic profile and saga: standardizing compute grid access in the open grid forum,” *Concurrency and Computation: Practice and Experience*, vol. 21, no. 8, pp. 1053–1068, 2009.

[33] I. Foster et al., “OGSA Basic Execution Service Version 1.0.” OGF, GFD-R-P.108, August 2007.

[34] D. Thain, T. Tannenbaum, and M. Livny, “Distributed computing in practice: the condor experience.” *Concurrency - Practice and Experience*, vol. 17, no. 2-4, pp. 323–356, 2005.

[35] D. Eager, J. Zahorjan, and E. D. Lozowska, “Speedup Versus Efficiency in Parallel Systems,” *IEEE Trans. Comput.*, vol. 38, no. 3, pp. 408–423, 1989.

[36] I. Foster, “There’s Grid in them thar Clouds,” personal blog, January 08, 2008, http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html.

[37] K. Keahey, R. Figueiredo, J. Fortes, T. Freeman, and M. Tsugawa, “Science Clouds: Early Experiences in Cloud Computing for Scientific Applications,” *First Workshop on Cloud Computing and its Applications (CCA’08)*, 2008.

Visualizing Conceptual Schemas with their Sources and Progress

Christian Kop
Applied Informatics
Alpen-Adria-Universitaet Klagenfurt
Klagenfurt, Austria
chris@ifit.uni-klu.ac.at

Abstract - **Conceptual modeling for database design is more than just a "drawing" of the database architecture which is readable for specialists. Instead it must be a means for communication between the database designers and the other stakeholders. Even the specialists are not only interested in the graphical representation. There is also a need that the database designers and end users get an overview if the focus of the database schema under development still reflects the expectations of the end users. Stakeholders are also interested in the current working state (progress) of the model. Therefore, it needs simple and easy to use techniques for gathering and presenting different kind of information. In this paper, a combination of such techniques is presented. Firstly, it will be proposed how a glossary based representation together with a graphical representation and a verbalization of concepts can be used for communication with the end user. In the remaining parts of this paper these techniques will be applied to give database designers an overview of the focus of the schema, the current progress state as well as an overview to the sources which are related to the model elements.**

Keywords - ***conceptual modeling; verbalization; glossary; progress information; important concept;***

I. INTRODUCTION

A database is the backbone of information systems. Therefore, conceptual database design is a very important aspect of information systems development. Wrong conceptual models can lead to serious problems since the software depends on the right concepts and correct relationships between these concepts. Later changes in the database design can lead to numerous changes in the information systems software or to unforeseen errors. Much effort must be spent on the communication and negotiation process with all the stakeholders to get a validated conceptual database schema. Thus, it would be good to work with a presentation technique that is easy to understand and as many stakeholders as possible feel comfortable when using such a technique. Unfortunately this is not possible because of the different skills and knowledge of the stakeholders. Some of them are domain experts with no knowledge in computer sciences, others have a little knowledge. The problem is even worse since it is also situation depended. Thus, a single representation technique that is perfect for all stakeholders does not exist. A solution could be a mixture of representation techniques. Hence, the success of database projects strongly depends on a good mixture to gather the information from the end users as well as to present this information to them.

The most commonly used representation of conceptual models is a graphical representation. Since the beginnings of conceptual modeling (i.e. entity relationship modeling) models were represented with a graphical language (e.g., entity types as rectangles). This has not changed over the time. Some parts of the Unified Modeling Language (UML) have still a graphical language (i.e. classes appear as rectangles, associations as lines etc.). However, over the time computer scientists got aware that such graphical languages are good for IT professionals but typical end users are not able to understand them. Therefore solutions to verbalize the conceptual schema were introduced. Verbalization means that the graphical language is transformed back into natural language descriptions. Beside the classical graphical representation and verbalization, in this paper it is proposed that in addition a glossary representation should be considered as a third possibility. All these three representation techniques together can help the stakeholders to understand the conceptual schema.

For computer scientists, there are still good reasons to use graphical modeling languages. They provide a good spatial overview over all the concepts and their relationships. Furthermore a graphical language with a well defined grammar and defined notions is better suited to generate a logical model for the database.

Natural language descriptions of a diagram can better explain concepts and their relationships. Finally, if glossaries are used as check lists, they can support the negotiation process. Using these three representation techniques together can compensate the weaknesses of a single representation technique. Hence, the best solution would be to have all the three representation techniques under one roof. This can give all the stakeholders the opportunity to read that representation which is the best for them in a certain situation.

Most of the tools for conceptual modeling are focused on the graphical view. Some tools and approaches only provide at most two main views (graphical view and natural language descriptions). Glossaries, natural language descriptions and diagrams together are not used in the context of database design, since most researchers rely on diagrams only.

Independent from the representation technique, the schema itself is only part of a greater design context. Every element within the schema must be traced back to a requirements source. During the design process, different elements in a schema will have a different working state. Whereas some elements are nearly completely modeled some elements still have to be finished. For some elements the designer must still ask questions or has open tasks in the task list, for other elements there are no more questions or tasks to do. There should be also the possibility to view concepts according to their importance in the schema. This is another kind of structuring mechanism to avoid that the stakeholders get lost within the network of concepts.

Therefore the paper is structured as follows. In Section 2, the related work is discussed. Section 3 gives an overview of two projects which were accomplished. Learning's of this project and the approval of previous research ideas and assumptions for the selection of the visualization strategies are presented in Section 4 namely graphical representation, verbalization and a glossary representation. Section 5 and 6 present additional visualization techniques based on the three basic visualization strategies. Section 7 shows parts of the tool. Section 8 summarizes this contribution.

II. RELATED WORK

Graphical representation (e.g., diagrams) is the most established type of representation for conceptual modeling in general and database modeling in particular. In the beginning of conceptual modeling, graphical languages like the Entity Relationship approach were proposed for both end users and database designers.

According to the underlying paradigm of how a stakeholder perceives the "world", two types of conceptual modeling approaches can be distinguished:

- Entity type and object oriented approaches,
- fact oriented approaches.

In the first paradigm the "world" is seen as a world of objects which have properties. Therefore a clear distinction is made between object and object types respectively and their properties. Representatives of this paradigm are the classical ER approach and UML. Fact oriented approaches on the other hand see the "world" as a world of facts. Facts describe objects and their roles within a relationship. No distinction is made between objects and properties. Every concept is treated equally. Representatives of this kind of paradigm are NIAM [14] and its successor ORM [8],[9]. Both approaches have pros and cons. Object oriented approaches look very compact. In a typical object oriented class diagram attributes are embedded in the class representation. No additional connections between classes and attributes are necessary which would expand the diagram. On the other hand, many revisions must be made if such a diagram is used too early in the design phase. Due to information that is collected, classes might become attributes and attributes might become classes. According to [8][9] this is a reason why fact oriented approaches are better suited for conceptual modeling.

Nowadays there are doubts that currently used graphical representations will support the communication between end users [13]. Therefore, it is proposed that more effort must be spent to produce good "diagrams" for user communication. Some researchers even state [4] [10] that the graphical representation of a conceptual model should be transformed back to natural language. In particular, they argue that this transformation better helps the end users to understand the very compact and sometimes formal graphical notation. As a solution for the transformation result, they often provide a restrictive form of natural language called controlled language [6]. Hence, the purpose of such a transformation (verbalization) step is to comment and explain the more formal graphical representation of relationships and concepts.

The use of glossaries and dictionaries was proposed since the 70. The first work on "glossaries" was done by Parnas [15]. He used tabular representations for the representations of functions. In the 80s the DATA ID approach [2] used glossaries as a central concept in their methodology. Requirements were distributed to data, operation- and event glossaries. The glossaries were the basis for traditional conceptual schema generation (ER diagrams and Petri nets). The KCPM approach [12] continues and extends this representation idea. It combines this idea with the fact oriented paradigm.

A similar technique to glossaries namely forms and templates were introduced for the description of use cases [3]. Another approach using form templates for functionality and navigation is NDT. It is described in [5]. In addition, the need for glossaries to describe also ontologies is proposed in [11].

Diagrams, verbalization strategies as well as glossaries can help to communicate with the stakeholders. Since the type of representation strongly depends on the skills of the stakeholder and the situation, a combination of all three representation techniques is always better as one representation alone. A lean modeling language which only consists of concepts and not of classes and attributes prevents that the database schema must undergo many changes.

Beside the communication to the end users it is also necessary that the designer knows the current working state within the model. Furthermore, he must know if each concept in the schema is related to at least one requirements source. Finally it would be good if he is supported in the question: "Do I still focus on the right things?"

Measures for the progress of requirements are given in [21]. These measures are based on the IEEE quality standards for requirements. Also in [20] an approach for measuring the progress of requirements was discussed. This approach mainly depends on the decomposition of requirements and the number of statements like "*to be completed*". However, this could lead to two problems:

- when to end with the decomposition and
- forgotten "*to be completed*" statements

As a consequence, in [22] measurement is based on templates and not on natural language requirements as described in the two other approaches. Particularly glossaries are used. With this strategy the "to be completed" statements become superfluous. Hence, there is no problem if the designer forgets them. Instead any gap (empty cell) in the glossary is a hint for missing information.

The best practice to visualize the relationships to requirement sources is a traceability matrix [24].

Related research results which can help to determine if the designer still focus on the right things were found in the area of schema clustering [16][17]. In this field so called centered entities are used as a basis for the clustering. Other ideas were presented in the domain of ontologies [18][19]. Key concepts were mainly used to give one measure for the quality of an ontology.

To summarize the related work: Different representation techniques are proposed in literature. However, usually only one technique or a combination of two techniques is proposed. This paper proposes to combine the three representation techniques, namely a glossary based representation with a graphical representation and a verbalization. Furthermore it proposes to use the combination of these three representation techniques not only for the schema itself but for a specific content aspect (i.e. important concept) as well as for context information (progress information, relationship to sources). Hence, it is the aim that the stakeholders get a holistic view on the database schema.

III. PRACTICAL EXPERIENCES

Before the approach of different visualization techniques is described, two real projects are presented in this section as an additional motivation to the literature study. The two projects were accomplished in two different domains. The first project dealt with the management of cancer cases. Each province has an appointment from the government, that a central institution should collect the appearance of cancer cases. These are used by the government for statistical analysis. Usually a central institution located in one of the public hospitals takes care of this. The order was to support this institution during requirements elicitation and analysis. The institution worked already with an information system for managing cancer cases since the nineties. However, as the reader can imagine, within ten years, knowledge about cancer cases has grown and requirements of managing data and especially statistical data about cancer has changed. Therefore it was necessary to develop a new system.

The second project is located in the area of electrical power plants (mainly hydroelectric power production). A central institution monitors all the plants in the province. It checks if all plants work correctly and it has to react if an accident happens (i.e. to assign a team to fix the problem) or the plant is switched off (e.g., because of maintenance). The crew which monitors all the plants has to note all the events so that there is a traceable logged documentation if there is a shift changeover of the crew. Also for the management it is interesting to see what is going on, which accidents happened and the reasons for switch offs. Although the monitoring crews have access to different data sources, they need a summary of all these information in a central database.

Beside their differences, both projects can be characterized by the following similarities:

- The projects had a strong data centric aspect. Data was needed to get statistical information and to support the decision making in both cases. Conceptual modeling to design the new database and communication with the stakeholders were important tasks.
- The project was not built from scratch. Either the data in the old system (cancer cases project) had to be considered or the new system has to gather and "summarize" data from different data sources (power plant project). However in both cases there was not such an amount of data that the development of a data warehouse was justifiable.
- In particular, it was also necessary and useful to analyze the type of data available in the old database or other data sources.
- In both projects the stakeholders agreed that a new system with new features is necessary. For the "cancer case" project, the old database system was outdated. Only those data which has proved to be interesting over all the years was kept together with new information that was needed because of the new knowledge. For the power plant systems the stakeholders needed a new database system which stores the integrated data from the different data sources.
- Because of the different skills, background and knowledge of the end user it was not possible to describe the needed data with class diagrams only.

Especially the last mentioned similarity underlines the proposals found in literature and was a motivation to think about a combination of three representation techniques and to apply these techniques also for specific purposes (i.e. progress information, relationship to sources). Since the two projects were data centric, the remainder of the paper focuses on visualization strategies for a conceptual database schema and will not discuss any other aspect of a software system (i.e. function, behavior, user interface, non functional requirements etc.)

IV. THE THREE VIEWS

A. *The model elements and graphical representation*

Before describing the several views the model is briefly introduced here. It is based on the ORM paradigm (facts instead of entity types). Therefore no distinction between classes and attributes is made.

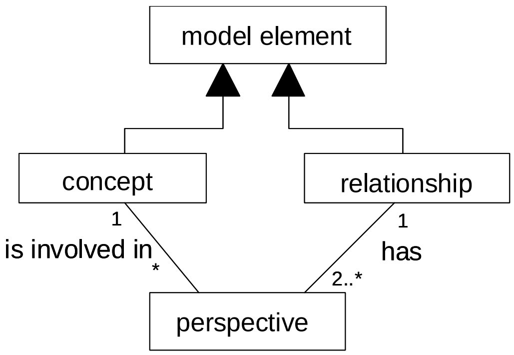


Figure 1: excerpt of the meta-model

The excerpt of the meta-model in Figure 1 illustrates this fact oriented paradigm. A concept is connected to a relationship via perspectives (roles in ORM). Both concept and relationship are model elements.

A concept itself is every term which is important in a certain domain. A concept can be a material or immaterial thing. It is also a term which would be modeled as an attribute in UML (e.g., first name). This supports the idea that designers shall elicit important concepts without thinking if they will become classes or attributes. Such distinction can be delegated to the tool.

Although the meta-model follows the fact oriented paradigm which allows that a relationship has more than 2 perspectives (e.g., ternary relationship) the representation of relationships is more similar to UML. Perspectives (roles) are hidden in the representation of a relationship. They are mainly used to specify the relationship. In this aspect it differs from ORM which strongly focus on roles also in the graphical representation. Especially special relationships (e.g., aggregation) are defined by pre-defined perspectives (e.g., aggregate_of, part of). Beside the well known relationships like aggregation, composition and generalization also an identification relationship and a hasProperty-relationship are part of the approach. The hasProperty-relationship which was introduced in [23] can be used to indicate that A has the property B. That B is a property of A does not necessarily mean that B is an attribute of A, if A and B would be mapped to an UML class diagram. B will only become an attribute if B does not have relationships to any other concept in the schema. On the other hand A can be transformed immediately to an UML class since it was specified with the hasProperty-relationship that A has a property. With this relationship alone a graph of concepts can already be easily transformed to an UML class diagram. The hasProperty-relationship is drawn with a directed edge pointing from the object representative to the representative of the property. Whereas the perspectives are predefined (“*has*”, “*belongs_to*”) the whole relationship can be labelled individually. The “*identifies*” relationship is used if the designer knows that the value of a concept identifies another concept. The predefined perspectives of this relationship are “*identifies*” and “*is-identified-by*”. The whole relationship is presented as an edge with two lines crossing the edge at the position of the identifier. With the two crossing lines, the relationship should appear like a “*key*”. The crossing lines represent the teeth of the key. The identify relationship must be used to model concepts (attributes) which will become key candidates in the database schema. If no special relationship is applicable, then also a (simple) binary relationship can be used with no special meaning. It is represented as an edge with no additional graphical features. The user freely can label the two perspectives as well as the whole relationship or leave the labels empty. Figure 2 shows the appearances of the different relationships.

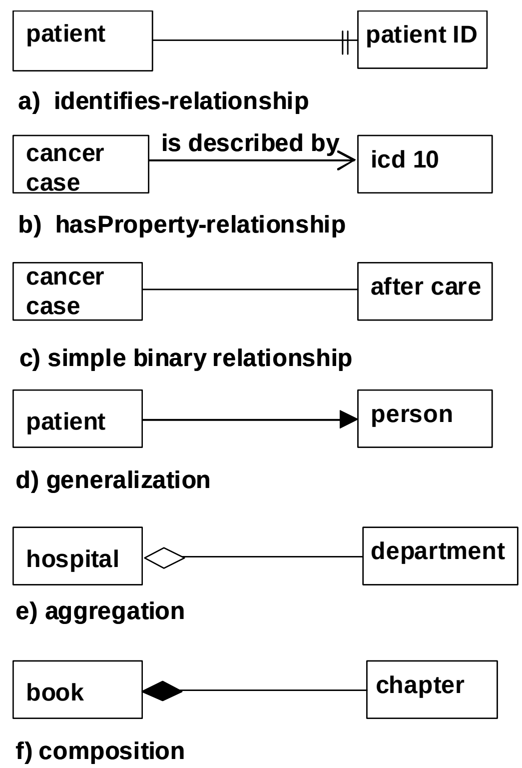


Figure 2: representation of relationships

Multiplicities must be defined for the normal binary relationship, the aggregation and for the *hasProperty*-relationship. There is no need to specify the multiplicity at the composite perspective of a composite relationship since the composite relationship has the same multiplicity semantic as the composite association in UML. There is also no need to specify multiplicities at all for the "*identifie*s-" and "*generalization*-"relationship. Because of their special semantics the multiplicities are implicitly defined.

Another difference to UML, ER diagrams and ORM is the management of additional concept information (e.g., examples, quantity descriptions, synonyms, value constraint). Since this information is well suited for a glossary view, it will be described in detail in the section which treats the glossary view.

Because of the semantic relationships, the information gathered in the glossaries (e.g., value constraint), information about multiplicities, concept name analysis and relationship name analysis, the approach allows an easy transformation to an Entity relationship or UML diagram. Hence, like ORM the approach is stable against changes in the model but can be transformed to UML. For a more detailed description of the model and the mapping, the reader is referred to [12] and [23].

B. Glossaries

Glossaries should not compete with the other representation techniques but try to complement them. Whereas graphical representations are good for a (spatial) overview and natural language descriptions explain formal notations to end users, the aim of a glossary should be a detailed but compact description of concepts. They should provide the negotiation process and also the process of collecting concepts from the stakeholders. Especially during the first stage of database development, a database designer is more like a medical doctor or a pilot who must work with check lists in order to get new information or validate old information. With glossaries the collected concepts will appear in a very compact format which is still readable and understandable for all the stakeholders.

Most often a concept glossary only has a column for the name of the concept and a column for the definition of that concept. With this kind of information a glossary would only play a minor role.

With additional glossary columns, different stakeholders can be incorporated (e.g., typical end users with no technical knowledge and persons with technical knowledge about the old system).

For instance, in the cancer case project there was a person who maintained the old system and of course the typical end users like physicians, nurses and secretaries. In the electrical power project, a project member from the customer's side had knowledge about the existing data sources from which the data should be extracted. If it is interesting in particular to represent information for all the stakeholders like it was in these projects, then such a glossary must not only consist of a concept name and definition column. Instead the following additional columns are necessary:

- Examples for the concepts,
- quantity description,
- synonyms,
- value constraint,
- data source constraint.

Figure 3 shows the part of the meta-model that manages a detailed description of concepts. It is visualized with the model elements presented here.

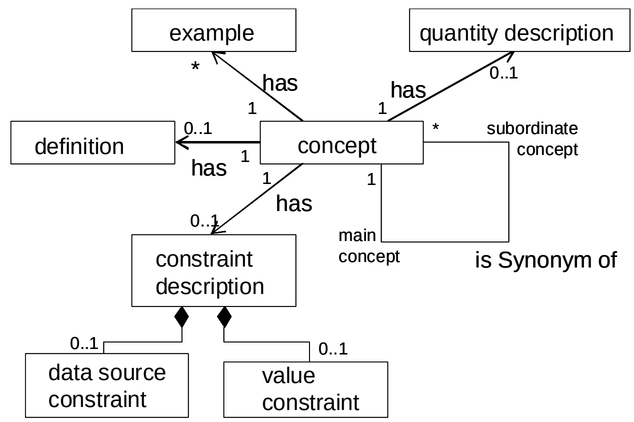


Figure 3: concept information

According to its name, instances and values are stored in the **example** column. (e.g., "*pathological institute*" for the concept "*department name*").

Quantity description specifies the amount of instances a concept will have (e.g., "*500 patients*"). It can be further refined with an indicator that tells if it is an average, a minimum or a maximum value (e.g., "*in average 500 patients*"). Additionally, with a second descriptor it can be specified that the quantity increases within a certain period (e.g., "*10 additional patients per year*").

Synonyms refer to other names of the same concept. (e.g., *institute* as a synonym for *department* if department was chosen as the main working concept). Usually synonyms have no internal hierarchy or ordering. If a notion N1 is synonym of a notion N2, then also N2 is synonym of N1. For conceptual modeling it is necessary to decide which concept will be further used. This is selected as the main concept in the list of synonymous concepts. The other concepts are still necessary but only in the sense that they represent variants of the main concept. Therefore in the synonym relationship of the meta-model the perspectives (roles) main concept and subordinate concept were introduced.

The **value constraint** consists of the sub information *format* and *data type*. The data type column specifies the data type a value can have. It can be a simple data type (String, Integer, Date etc.) or an Enumeration. It is intended that also smart business objects can appear in this column. Smart business objects as proposed in [7] are specific data types which are restricted to a certain format and specific features and operations (e.g., a type "*e-Mail address*"). In the format column, the appearance of the values is specified. The simplest form is the definition of the length of a value (e.g., a String or an Integer value). The length is encoded with the character L. Thus, a string value with 50 characters is encoded with L50 and has the entry “String” in the data type column. If a concept like diagnosis date is based on the data type “DATE”, then this data type appears in the data type column. If such a data has a specific format (e.g. “YYYY/MM/DD”) then this is collected in the format column.

The **data source constraint** consists of the same information. In addition it has the column *data source*. The *data source* constraint column was introduced since often an old database system exists which has to be replaced by a new one. It cannot be expected, that the data types and formats will stay the same in the new version of the system. If data from the old system is migrated into the new system, possible differences of data types and formats must be considered in advance. The additional column “data source” specifies the source and the structure of a concept in the old system from which data has to be migrated. The expression “<table name>.<attribute>” is used for it. If the concept is only a table in the old database, then only “<table name>” can be used. Table 1 in the appendix shows how such a representation can look like.

C. Verbalization

For verbalization it is assumed, that class names and concept names respectively are in singular form. Relationship names are verbs in 3rd person singular form. These verbs can be either given in active or passive voice.

Usually a verbalization of a diagram is made by paraphrasing the graphical content. In particular, the approach described in [10] uses the label of object types, the labels of the roles and the multiplicity information. This information is concatenated together with fillers (e.g., articles, quantifiers) to form a natural language sentence. Especially the multiplicity information must be translated from a number representation (e.g., numbers in brackets [0..1]) to its textual representations. Short cuts like “*exactly 2*” for [2..2] must be considered.

As mentioned above a concept name is written in singular in the graphical representation. In the resulting sentence of a verbalization, it can be left in singular if it is the subject of the sentence. If it is the object in the sentence then it has to be decided if this concept name must be transformed to the plural form. The decision is based on the multiplicity information (e.g., N as the maximum multiplicity).

The verbalization strategy also takes the special relationships between the concepts into account. Beside the commonly used special relationships like “*Generalization*” and “*Aggregation*”, the model also offers the special relationships “*hasProperty*” and “*Identification*”. These additional special relationships make it easier to verbalize the graphical representation of the relationship. In the “*hasProperty*” relationship the verbs *has*/*belongs_to* are taken as default paraphrases for the relationships between the concepts, but the designer always can use another word (e.g., *owns*, *buys* etc.) instead of *has*. In this case the word with which the user defines the relationship is taken in the verbalization step. The special relationship “*Identification*” provides two roles. These roles are *thing identifies (another) thing* and *(another) thing is identified by thing*. They are taken for verbalization. The place holder “*thing*” is replaced by the concrete involved concepts of the identification-relationship. If no special relationships are used, then it is recommended, that the user specifies the name of the relationship. Otherwise, the relationship is verbalized into a simple “*is related to*” phrase.

In addition to relationship verbalization, also a verbalization of some of the concept columns (quantity description column, format column and value constraint column) is provided. Special sentence pattern are used. A sentence pattern like “*There are [in average | at least | at most] <quantity> [additional] <concepts> [per year]*” can already support the verbalization of a quantity description. The phrases in square brackets are optional. The minimal specification of a quantity is “*There are <quantity> <concepts>*”. This is equivalent to “*There are in average <quantity> <concepts>*.” If there is an upper limit that can be reached, then “*at most*” is taken. If the quantity will never fall under a minimal limit then “*at least*” is used. If not the total quantity is meant but a quantity that rises per year then “*additional*” together with “*per year*” is added.

To specify the format column a sentence like “*The format of the <concept> is <format description>*” can be used. If enumerations are defined in the value constraint column of a concept a sentence like “*<concept> must be either <value> or <value> or <value>* …” is generated.

V. VISUALIZING IMPORTANT CONCEPTS

In the last section the three basic presentation strategies were introduced. This section builds on the three presentation strategies. They are used to visualize important concepts. Information about important concepts is useful to get a quick overview of the schema focus. Especially the two questions are of interest:

- Is the focus of the schema still the focus which was expected by all the stakeholders at the beginning of a database design project?
- Is a certain concept specified enough?

These questions can be broken down to the question of important concepts within a schema. If the important concepts modeled in a schema are not the same as expected by the stakeholders, then it is possible to detect a defect. For instance, such a situation can appear if an important concept is still underspecified. This can happen due to a misunderstanding between the designer and end users. Particularly, the designer concentrates on the description of concepts which are not so important for the stakeholders.

In order to get this information, it is necessary that the tool itself can automatically determine important concepts on the basis of already modeled information. An adjustment can then be made between the generated proposal of the tool and the expectations of the stakeholders.

This section will discuss this topic. After defining what important concepts are and how they can be calculated, it will be explained how the different views can visualize this kind of information.

A. What are important concepts?

The notion "important concept" is based on the idea, that they are well described in a conceptual schema. They make up the centers of the schema and other concepts (supporting concepts) are used to describe them. Synonymous notions for important concept are "centered entity" and "key concept".

[16] has introduced the notion "centered entity" for using it as a basis for a clustering algorithm. Entities are described in terms of relationships in which they are involved. Hence, the more an entity has connections to other entities; the more the entity can be seen as a centered entity. This definition of a centered entity is very pragmatic, based on the analysis of a graph. It has the advantage that it can be done automatically by the tool [16], [17].

Most important for the approach introduced here is a research result achieved by the same author some years later [17]. A study with students was made. One part of the study focused on the centered entities itself. The question was examined, if entities with more relationships are perceived as more important. The study showed that this is the case.

In the research area of ontologies the notion "key concept" was introduced in [19]. It was part of an approach which checked the quality of an ontology. Once again relationships were used for the calculation of key concepts. Here, the relationships are weighted higher, if more implicit relationships in the lower sections of the generalization hierarchy can be derived from them. In [18] only the children of a concept in a specialization hierarchy were counted.

B. How to calculate importance?

Since database design is more focused on relationships between concepts than on a generalization/specialization taxonomy, this approach follows the idea of [16][17]. It differs and extends this previous approach since it considers the type of relationship between the concepts. The calculation consists of two sub steps:

- Counting of connections to other concepts
- Categorizing a concept.

Counting step: For the approach presented here, the counting is done as follows: For each binary undirected relationship a concept has, the counter is increased by 1. For each special generalization relationship a concept is involved, the counter of that concept is increased by 1. If a concept is involved in the special aggregation relationship then the counter is increased only if it has the aggregation role. This is based on the idea, that aggregates more represent the main concepts than their parts since otherwise it would not be necessary to model the aggregate but it can be concentrated on the parts only. The two additional semantic relationships *identification* and *hasProperty* are also counted differently. The counter is increased for a concept if it is identified by another concept. The counter is not increased for the concept which identifies, since this concept can be understood as a (database key candidate) attribute. The *hasProperty* relationship is a directed relationship between a concept and its property representation (once again a concept). For each *hasProperty* relationship where the concept is in the role to have the property and not in the role to be the property the counter is increased by 1.

Counting in other approaches (UML, ER, ORM): For the sake of completeness, the step is also explained for UML, ER and ORM schemas. The counting of importance depends on the paradigm which is used.

For UML or ER the counting could be as follows: All attributes in an UML class diagram or Entity Relationship diagram get the count value 1. For each class, entity type respectively, their numbers of attributes are counted. For instance, if a certain class (entity type) has 12 attributes, then its initial count result is 12. For each binary (n-ary) undirected association, a class / entity type is involved in; the count result is increased by 1 for that class / entity type. For each generalization relationship a class / entity type is involved in, the count result is increased by 1. For each aggregation- or composition relationship a class / entity type is involved in as an aggregate the count result for that class / entity type is increased by 1. UML class diagrams provide two additional features, which are interesting for counting. Associations can be extended with a reading and navigation direction. In these two cases the count result is increased by 1 only for those classes which are the source (starting point) for the reading or navigation direction. It can be argued that the source of the navigation or reading direction is focused. Hence, it is more likely that it is an important concept than the target of the directed association.

For an ORM diagram the counting is as follows: For each role of an object type, the counter is increased by one. If the object type in addition has a key attribute, then the counter is increased by 1 once more. Aggregation is treated in the same way as shown for UML.

Categorization step: The result of the counting for each concept is now taken as an input for the categorization step. Additionally the concept with the maximum counting result is selected out of the list of concepts. This is the first detected important concept. The counting results of all other concepts are compared with this maximum.

The comparison returns to which category a concept belongs. The approach is restricted to the three categories:

v*ery important concepts*, *important concepts* and *unimportant (supporting) concepts*.

The distinction into which category a concept falls is determined by the percentage of counted connections a concept has with respect to the maximum counting result in a certain schema. If a concept reaches a percentage value >= 66 % then it is a very important concept. If the percentage value PV is 33 % <= PV < 66 % then it is an important concept. Finally, if the percentage value is below 33 % then it is only a supporting (unimportant) concept.

Let a concept C_1 have a count of 50, meaning it was able to increase the counter by 1 for 50 relationships it is involved in. Let us further assume 50 is also the maximum counting result that appears in this schema S. Let another concept C_2 in S have a counting result of 20. The type of the concept is then calculated by 20 / 50 and the result is 0.4 (40 %). With this 40 % the concept belongs to the category of important concepts. Let a third concept C_3 have the value of 40 which means, it reaches the maximum with 80 %. Hence, C_1 and C_3 belong to the very important concepts.

After this introduction what main concepts are and how they can be detected in the schema, the next section discusses how such information can be offered to the user in the different representation techniques presented in this paper.

C. Visualization

For the **graphical representation** a strategy was chosen, which is a combination of enlarging the rectangular dimensions of a concept together with a coloring strategy. The very important concepts appear as the biggest concepts on the screen. The color of this concept is deep green which emphasize their importance. Important concepts are also enlarged but not as much as very important concepts. They appear in a yellow color. This gives them a more transparent and pale touch. The color and the size signalize that they must be considered as important, but they are not among the most important. Finally the supporting concepts are not enlarged at all, but appear as they are. They have a white color, which underlines their supporting character. The spatial information is not distorted as it is only necessary to show which concept is very important, important or unimportant.

In the **verbalization** view all unnecessary information is filtered out to avoid textual bulk. Like in a news paper, book chapter or any other linear textual description an abstract or summary of what has been specified is provided to the reader. For those kinds of concepts which are important according to their specifications the user gets a very detailed and insight look. On the other hand he will not be bothered with details of supporting concepts. They only appear in the textual summary as long as they help to describe at least one of the (very) important concepts. Such a verbalization can start with a textual introduction template like: "*The most important concepts of this schema are <list of very important concepts> followed by <list of important concepts>*". Afterwards each of the (very) important concepts is verbalized according to the strategies described in the verbalization section.

The **glossary content** can be sorted. For sorting, an additional glossary column is introduced. In this column, the counting results are presented. If the glossary rows are sorted according to these columns in a descending order, then the very important concepts appear before the important concepts and the supporting concepts.

To summarize, if for instance "*after care*" is seen as an important concept in the medical (cancer) domain then such visualization strategies can help to detect a defect. For this example, the reader is referred to Figures 5 and 8. In Figure 8, "*after care*" is only presented as an unimportant concept. In the textual summary it only appears in the description of *cancer case* but is not itself described. In a glossary representation it will not be among the first listed concepts. Hence, if this concept is important for the stakeholders they will be surprised on one hand but on the other hand they will get aware that something (i.e. a better description of *after care*) is missing.

VI. VISUALIZING THE PROGRESS AND SOURCE

Up to now visualization of model elements were described only. In fact a concept is not only related to another concept but it is also "related" to sources from which it was derived and it is related to a certain working state (progress). If the stakeholders need a holistic view of the model, then also their relationships to the sources and the working progress of the model is information that must be visualized appropriately. Figure 4 shows the relationships between a model element to its sources and its progress information.

In this section it will be firstly defined what is meant with source and progress. Then it will be explained how such information can be visualized.

A. Source

A source is any thing or media from which a model element like a concept can be derived.

In this approach three kinds of sources are distinguished

- natural languages requirement sentences,
- documents
- involved persons

A natural language requirement sentence is the smallest unit of source from which a model element (here a concept) can be derived. The requirement sentence itself can be selected from a document.

A document is any type of media in which a model element was found.

An involved person is any stakeholder who mentioned the model element.

Instead of using very small units of single requirements sentences only, this approach also allows to relate a model element to the more coarse grained sources “document” and “involved person”.

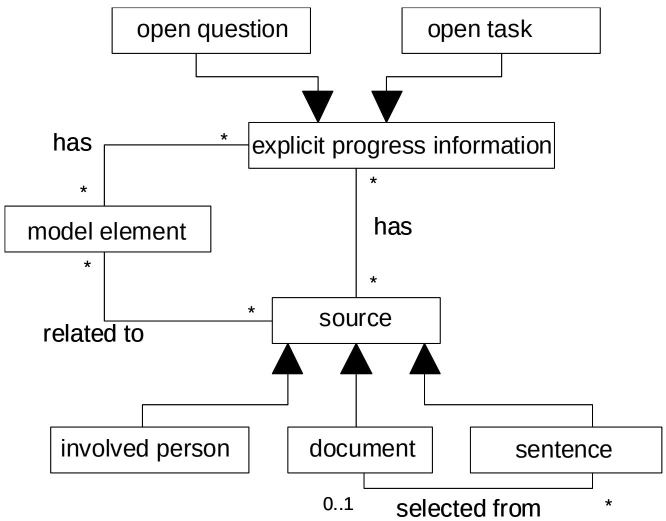


Figure 4: model element, progress and source

B. Progress

It might be surprising that progress, which is a certain state or snapshot of the modeling process, is specified in the meta-model. However “progress” can be divided into **explicit progress information** and **implicit progress information**. In this approach an explicit progress state is given if the designer declares that there are still some open questions or open tasks for a certain model element or a source. An implicit progress cannot be declared explicitly but is derived from the grade of completeness of the schema. Therefore the meta-model only specifies the explicit working progress. Nevertheless, the details of implicit progress information are also given here.

The **explicit progress information** (open questions and open tasks) is necessary for the following reasons:

- Conceptual modeling is always driven by decisions (i.e. decision to model certain information in a certain way, decisions to select and gather some information whereas other information is ignored etc.). Some of them can be made by the designer itself whereas other decisions need communication with end users. If the designer is not sure if he has made the right decision, then a possibility must exist to make a remark for asking the end user. Furthermore this remark must be related to the respective model element. Such a remark is not only a hint for the designer to ask somebody something, but also a concrete hint that the element is not yet finished.
- Not every task can be done at once. Some tasks must be done later. The open task remark helps the designer to remember these tasks (e.g., “*I must collected detailed information for the concept patient*”). Once again this remark is a concrete hint that something is not yet finished.

The **implicit progress information** can be derived by answering the following two questions:

- Is each column in the glossary view filled with a value?
- Is each concept related to at least one other concept and is the multiplicity information within each relationship specified for a certain concept.

The answers for the first question can be found inside the schema structure itself. In [22] a general method to calculate the progress was already introduced. Therefore this paper concentrates on the visualization part which was not the scope in [22]. The customization of the general method is only explained to the extend that is necessary to understand the visualization. Imagine a matrix similar to table 1 in the appendix. Each row describes a concept. Each column is reserved for a specific aspect of a concept (e.g., its examples, its definition, its value constraint, its quantity description – see Figure 3). A cell of a certain row and column can be empty or filled. The concept definition and the example column are two kind of information that must be filled out in order to be complete. The progress can be calculated by counting only those cells of columns which are filled with a value and dividing them by the total number of columns which must be filled out. If the total number of columns would be 2 (i.e. concept definition column and example column) and if only one is filled out for a certain concept at a certain point in time, then the progress state would be 50 %.

Additionally, the approach also considers the columns for value constraint and quantity descriptions. For the calculation of the progress state of these concept aspects, the general method described in [22] was customized and refined. The quantity description is not needed for every concept. Therefore it is optional. However, if a quantity (numeric value) is specified for a concept, then all the other information must also be specified (e.g., *average/maximum/minimum*, i*ncreasing per period or not*). The data type entry in the value constraint depends on the state of the concept. If the concept is already categorized as an attribute, then the data type must be filled out. If it is a class then it must not be filled out. If it is a concept which is not yet categorized to a class or an attribute and if the data type is empty, then the implicit progress information for that concept is defined as 0.5. This gives a hint that there might be still something missing.

Consider a concept which is a class. It has a quantity description and all the necessary information for this quantity description is specified. A concept definition is specified but no examples are given. In this case the progress state is 2 / 3 (~ 66 %). On the other hand if it is an attribute that has a data type entry but no (concept) definition and no examples then the state is 1 / 3 (~ 33 %). Finally if there is a concept which is not already categorized as a class or an attribute and it has examples and a concept definition but no value constraint then the progress is calculated as 2 / 2.5 (~ 80 %).

The answer of the second question can be calculated by determining if a concept has at least one relationship to another concept. Then for every relationship which does not have predefined multiplicities (e.g., "*identifies*" relationship) the multiplicity information of the concept to its related concept is examined. The state of completeness for relationships and multiplicity information is defined as follows:

- If the concept has no relationship then the implicit progress information for the relationship progress state (RP) is 0. This means incomplete.
- If the concept has at least one relationship to other concepts, then the relationship progress state (RP) is determined by:

$$RP = \frac{\text{No. of specified multiplicities}}{\text{No. of relationships for a concept}}$$

For example, if the concept *cancer case* has 10 relationships to other concepts (e.g., *start location*, *histology description* etc.) but only for 4 of these relationships the multiplicities to the other concepts are defined, then RP = 0.4 (40 %).

The whole implicit progress information is calculated by building a sum of RP and the other progress state information (e.g., progress of example, concept definitions etc.). This is then divided by the possible number of progress information. The result is the overall progress state in percentage. As a continuation of the previous example, let us assume that cancer case would have an RP of 40 %. Furthermore it is categorized as a class and has examples and a concept definition. In this case the whole implicit progress state is 80 %. If all the relationships also have specified multiplicity information then the progress state is 100 %.

Implicit and explicit progress information is visualized separately, because situations can occur where a schema is already finished according to the implicit progress information, but it is not finished according to the explicit progress information. An example for such a situation is the following: The designer has already filled out and modeled the necessary information but in one case he is not quite sure if his design decision is correct. Since he has to ask one of the end users, he makes a note (open question) to ask somebody. In other words, from a structural point of view a certain model element in the schema is complete but it is not yet validated by the end user.

C. How to view the Progress information

There are several ways to **graphically** view the implicit progress information. One is to resize the concept. The more information about a concept reaches the state "complete", the bigger it appears in the graphical view. Alternatively the more a concept is not completed, the bigger it could appear in the graphical view. As a third possibility the concepts can appear in the different colors of a traffic light. The semantics of the colors are:

- *green*: concept is largely specified or even complete (>= 66 %);
- *yellow*: concept needs more information (>= 33 %)
- *red*: concept is barely specified (< 33 %)

It was decided to use this third possibility. For instance if the first alternative would have been chosen, then incomplete concepts only appear very small although the focus of the users attention should be directed to these incomplete concepts. On the other hand, if incomplete concepts are drawn very large then the mistake can occur that these concepts are seen as complete concepts.

Explicit progress information is graphically visualized in the same way:

- *red*: there is at least one open question or one open task respectively.
- *green*: no open questions; no open tasks.

Glossaries itself are a good view to visualize on a very detailed level that something is missing, since in this case the cell of a row and column is empty. If an overview of the progress is needed, then this can be achieved by a table consisting of four columns. The first column contains the concept names in each row. The second column contains the progress of this concept using a progress bar (see Figure 6). In the third column each cell is colored green if no open question is stored for a concept. If at least one open question exists, then the cell has a red color. In the fourth column the same visualization strategy is applied for open tasks.

A good strategy for verbalizing the progress was not found. Of course, there is always the possibility to verbalize the percentage of progress for each concept or to name the columns of a concept which are not filled out. However, glossaries or a graphical view are much better in such a situation since verbalization is a strategy which presents content itself and not the gaps.

D. How to view the relationship to the sources

In the same way as the amount of relationships to other concepts is visualized graphically, the strategy can be applied to visualize the relationship to sources. However, it cannot be concluded from such a counting strategy, that a concept with more relationships to sources is more important than a concept with fewer relationships. It might happen that a concept was only (but completely) found in one document or was specified by a single person. Hence, if in the graphical view a concept appears in a bigger size it only tells, that it has more relationships to different sources.

In the glossary view, the relationship can be viewed with the already well established strategy of a traceability matrix. In its most general form, there is one column for the concepts and columns for each kind of source (involved person, document and sentence). In the cell where a column and a row cross, a number indicates to how many sources of a certain kind (e.g., document) a concept is related. Once again no adequate strategy was found actually for the verbalization view.

VII. THE TOOL

A. *General Views*

A tool (see Figure 7 in the appendix) was implemented to meet the requirement that verbalization, graphical representation and a glossary view must be combined.

The left upper part of the tool presented in Figure 7 is the verbalization view. Here the diagram appears as a description written in controlled language. These sentences are generated from the information of all the relationships and concepts. Relationship information include involved concept names, the name of the relationship (e.g., *is a*, *identifies*, *has*, *owns* etc.), and multiplicity constraints. Concept information is information about the value constraint, the format and the quantity description specified for a certain concept.

The right upper part of the tool is dedicated to the graphical representation. This is the classical form of representation used in conceptual modeling languages.

At the bottom the set of modeled concepts appear in a glossary style. The user has the advantage to use the list of concepts like a check list. He can look which columns are filled out and which are empty.

To ensure that the user will not be overburdened with three different views, of course it is possible for him to switch off one view completely. The user can also resize the different views to get a larger glossary view, a larger graphical view or a larger textual view.

Currently the model elements can only edited in the graphical view. The textual view offers only the possibility to insert controlled language sentences or read these sentences from a file. With a button in the text view panel, these sentences can then be transferred to the graphical view.

B. *Visualization of Important Concepts*

The visualization of important concepts currently is implemented in the following way. For the graphical view of important concepts, the designer has to click on the button with the "spyglass" icon. Then he gets a popup window with a menu of several possibilities. One option is to choose the visualization of important concepts. After he has selected this option, important concepts appear as described in three different sizes and colors (see Figure 8 in the appendix). If he wants to see a natural language summary of the important concepts, then he must select the tool menu option "Views" in the menu bar. Afterwards he must select the submenu item "Summary". Finally a window is popped up and displays the textual summary (see Figure 5). If he wants to see the glossary view then he also has to select the "Views" menu. Finally he must select the sub menu "Important concepts listing". A window is popped up where the concepts are ordered according to their importance.

C. *Visualization of progress and sources*

The graphical representation of the progress of concepts and their relationships to sources can be reached through the button with the "spyglass" icon in the graphic panel. The designer must then chose the corresponding option, depending of what he wants to see:

- Visualization of explicit and implicit progress information
- Visualization of relationships to sources

The right upper graphical part of the tool gives the required view as described (i.e. traffic light coloring paradigm for explicit and implicit progress information; three sizes and colors for concepts to visualize the number of relationships to sources).

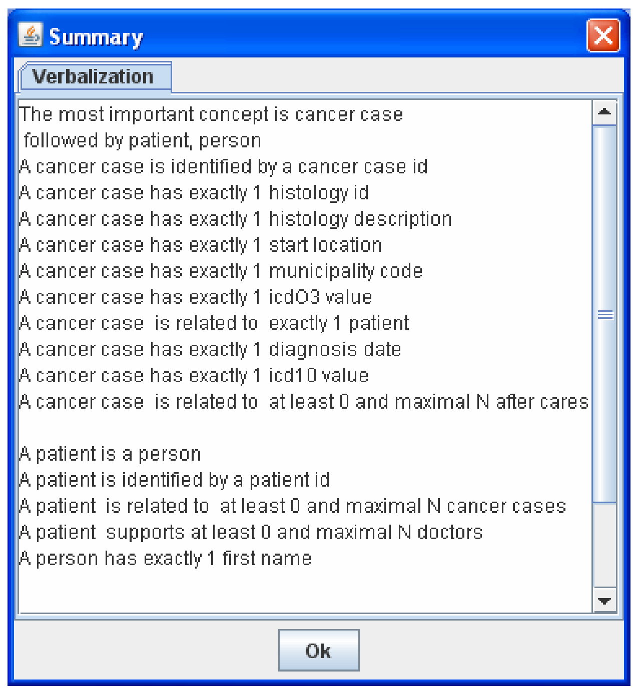


Figure 5: summary report

In order to get a glossary representation of progress information, the designer must navigate from the menu bar item "Views" to the sub menu "Progress information" and "Traceability overview" respectively. For each of the two options, a window is popped up which contains the necessary information (see Figure 6 for implicit and explicit progress information).

Verbalization strategies of progress information and sources are not supported at the moment.

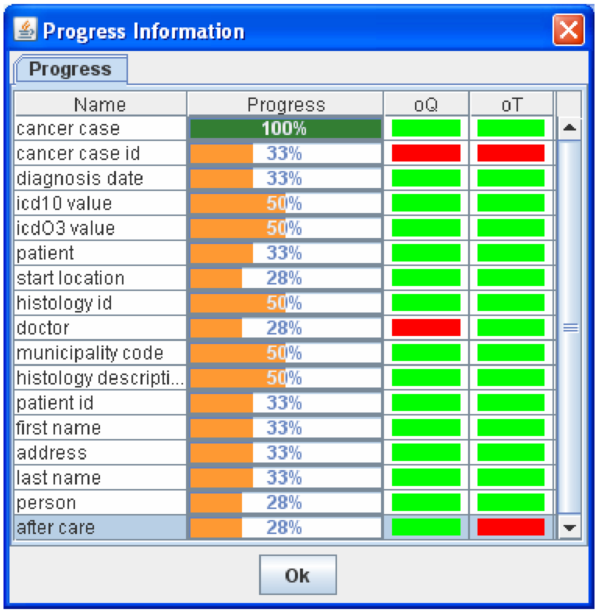


Figure 6: progress information

D. Technical aspects

The tool was implemented in Java and all the information of the concepts and their relationships are stored in a MySQL database.

The Model-View-Controller (MVC) architecture was used to manage the changes between the graphical and the glossary view. Inserting and updating of concepts and relationships is done in the graphical view. The user gets a property window for relationships and concepts. In these property pop up window he can insert and edit the details. Whenever details of a concept are changed then the graphical and glossary view is notified.

The verbalization is not trigged by these changes since the textual area in the left upper part of the tool (Figure 7) is also used as a simple editor for inserting a list of controlled English sentences which can then be transferred into the graphical and glossary representation. Instead a button in this area generates the verbalization from the model. The verbalization strategy itself is implemented within the MVC model classes for concept and relationship. Each of these classes has a public method "verbalize". Hence each object of these classes knows how to describe itself textually. The verbalization process itself is simply implemented by going through the entire concepts and relationships in a domain and by calling their method "verbalize".

VIII. CONCLUSION AND FUTURE WORK

It is very important that the result of conceptual modeling is negotiated with all the stakeholders. Since the stakeholders have different skills and knowledge background, different representation techniques should be used for the communication.

In this paper three representation techniques were combined to give all stakeholders the possibility to choose the most adequate one in a given situation. These visualization strategies were then applied to "structure" the schema between important and non important concepts in order to detect defects in the schema.

Since a schema must be seen in a greater context not only the visualization of the schema itself is relevant, but also the relationships of certain model elements to their sources as well as their actual progress of design. Both, overview of relationships to sources and progress information can help stakeholders to get a better picture about the current conceptual modeling state.

These strategies are based on previous research results, a survey of the literature and learning's made in projects.

In future, more special relationships might be added to this approach. Further special representation techniques for special purposes together with the existing techniques (i.e. progress or relationship to sources) should be studied for their optimal usability.

REFERENCES

[1] Ch. Kop, "Towards a Combination of Three Representation Techniques for Conceptual Data Modeling", First International Confernce on Advances in Databases, Knowledge, and Data Applications, 2009, pp.95-100.

[2] S. Ceri, (Ed.) *Methodology and Tools for Database Design*, North Holland Publ. Comp., 1983.

[3] A. Cockburn, *Writing Effective Use Cases.* Addison Wesley Publ. Comp., 2000.

[4] H. Dalianis, "A method for validating a conceptual model by natural language discourse generation". In P. Loucopoulos (Eds.), *Proceedngs of the Fourth International Conference CAiSE'92 on Advanced Information Systems Enginering*. Lecture Notes in Computer Sciences (LNCS) Vol. 594, Springer Verlag, pp. 425-444.

[5] M. Jose Escalona, G. Aragon, NDT. A Model-Driven Approach for Web Requirements IEEE Transactions on Software Engineering, Vol. 34, No. 3, 2008 pp. 377 - 390.

[6] N.E. Fuchs, S. Höfler, K. Kaljurand, F. Rinaldi and G. Schneider, "Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines" In Norbert Eisinger N. and Maluszynski, J. (eds.): *Reasoning Web, First International Summer School 2005*, Lecture Notes in Computer Science (LNCS) Vol. 3564, Springer Verlag, 2005 pp. 213-250.

[7] X. Liang, and A. Ginige, "Smart Business Object - A New Approach to Model Business Objects for Web Applications", In *Proceedings of the first international Conference on Software and Data Technologies (ICSOFT 2006)*, Setúbal Portugal 2006, Springer Verlag, pp. 30-39.

[8] T. Halpin, 'UML Data Models from an ORM Perspective-Part 1', *Journal of Conceptual Modeling*, No. 1, 1998, www.orm.net.

[9] T. Halpin, A. Bloesch, "Data modeling in UML and ORM: a comparison", *Journal of Database Management*, 10 (4), 1999, 4 - 13.

[10] T. Halpin, M. Curland, "Automated Verbalization for ORM 2", In *Proceedings, OTM 2006 Workshops -On the Move to Meaningful Internet Systems 2006*, Lecture Notes in Computer Science (LNCS 4278), Springer Verlag, pp. 1181 – 1190.

[11] M. Jarrar, "Towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering" In *Proceedings of the*

15th International World Wide Web Conference (WWW2006). Edinburgh, Scotland, ACM Press, pp. 497-503.

[12] H.C. Mayr, C. Kop, “A User Centered Approach to Requirements Modeling”, *Proc. Modellierung 2002*, Lecture Notes in Informatics LNI p-12, GI-Edition, 2002, pp. 75-86.

[13] D. Moody, “What Makes a Good Diagram? Improving the Cognitive Effectiveness of Diagrams in IS Development”. In. G. Magyar, G. Knapp, W. Wojtkowski, W.G. Wojtkowski, J. Zupancic (Eds), Advances in Information Systems Development – New Methods and Practice for the Networked Society, Vol.2, Springer Verlag, 2007, pp. 481-492.

[14] G.M. Njjssen, T.A. Halpin, *Conceptual Schema and Relational Database Design – A fact oriented approach*. Prentice Hall Publ. Comp. 1989.

[15] J. Ryszard, D.L. Parnas, J. Zucker “Tabular Representations in Relational Documents”, In Hoffman D., Weiss D.M. (Eds.) *Software Fundamentials – Collected Papers by David Parnas*. Addison Wesley Publishing Comp. 2001, pp. 71- 85.

[16] D.L. Moody, A., Flitman, “A Methodology for Clustering Entity Relationship Models – A Human Information Processing Approach”, In. Proceedings of Conceptual Modeling (ER 1999), Lecture Notes in Computer Science (LNCS), Vol. 1728, 1999, Springer Verlag, Berlin, Heidelberg, pp. 114-130.

[17] D.L. Moody., “Entity Connectivity vs. Hierarchical Levelling as a Basis for Data Model Clustering: An Experimental Analysis” In DEXA 2003 Proceedings, Lecture Notes in Computer Science (LNCS), Vol. 2736, 2003, Springer Verlag, Berlin, Heidelberg, pp. 77-87.

[18] D. Bezerra, A. Costa, K. Okada, *SwTOI (Software Test Onotlogy Integrated) and its applicaton in Linux Test*. In Proceedings of the 3rd International Workshop on Ontology, Conceptualization for Information Systems, Software Engineering and Service Science, CEUR-WS, Vol 460, http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/, pp. 25 – 36.

[19] N. Huang, Sh. Diao, “Structure-Based Ontology Evaluation” In IEEE International Conference on e-Business Engineering (ICEBE06), pp. 1- 6.

[20] R.J. Costello, D.-B. Liu, “Metrics for Requirements Engineeing, in Journal of Systems and Software, 1995, pp. 39 – 63.

[21] Ch. Pikalek „Messbare Qualität von Anforderungsdokument“, Javamagazin, No. 1, 2006, pp. 75 – 81.

[22] Ch. Kop, ”Work Progress Estimation from Structured Requirements Specifications“, In Ch. Barry, K. Conboy, M. Lang, G. Wojtkowski, W. Wojtkowski (eds.). Information Systems Development, Springer Verlag, Vol. 2, 2009, pp. 909 – 922.

[23] Ch. Kop, “Conceptual modeling tool for novice designers”, In International Journal of Metadata, Semantics and Ontologies, Vol. 3(2), April 2008, pp. 151 – 165.

[24] G. Kotoyna, I. Sommerville, *Requirements Engineering – Processes and Techniques*,Wiley Publ. Comp. 1998.

APPENDIX

TABLE 1 excerpt from the concept glossary

Concept name	Format	Datatype	DataSource	SrcFormat	SrcDatatype
cancer case			CCSTD		
cancer case id	L5	Number	CCSTD.ID	L5	Dezimal(5)
diagnosis date	YYYY/MM/DD	Date	CCSTD.DDATE	L10	CHAR(10)
icd10 value	L5	String			
icdO3 value	L5	String			
starting location		{left, right, unknown }			
histology id	L5	String			
histology description		Text			
patient			PSTD		
patient id	L6	Number	PSTD.PNUMBER	L5	Dezimal(5)
first name	L30	String	PSTD.FNAME	L21	CHAR(21)
last name	L30	String	PSTD.LNAME	L21	CHAR(21)
address	L255	String	PSTD.ADDRESS	L150	CHAR(150)
municipality code	L5	Number			
doctor					
person					

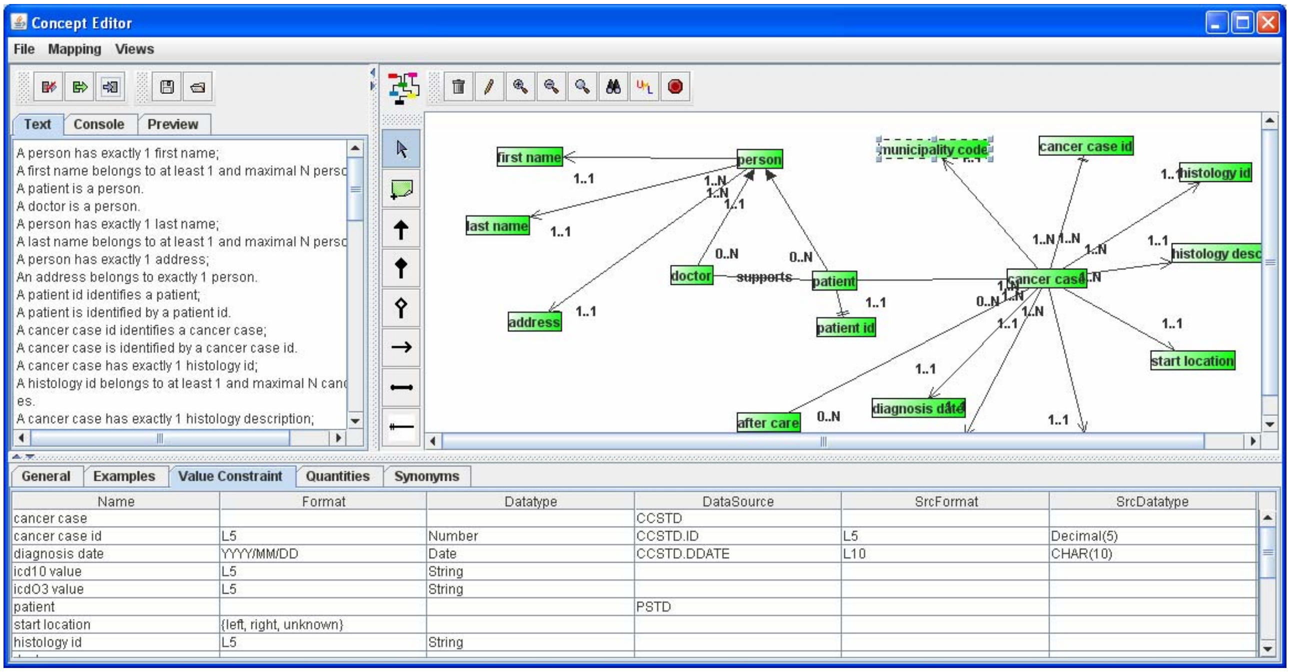


Figure 7: tool with the three presentation views

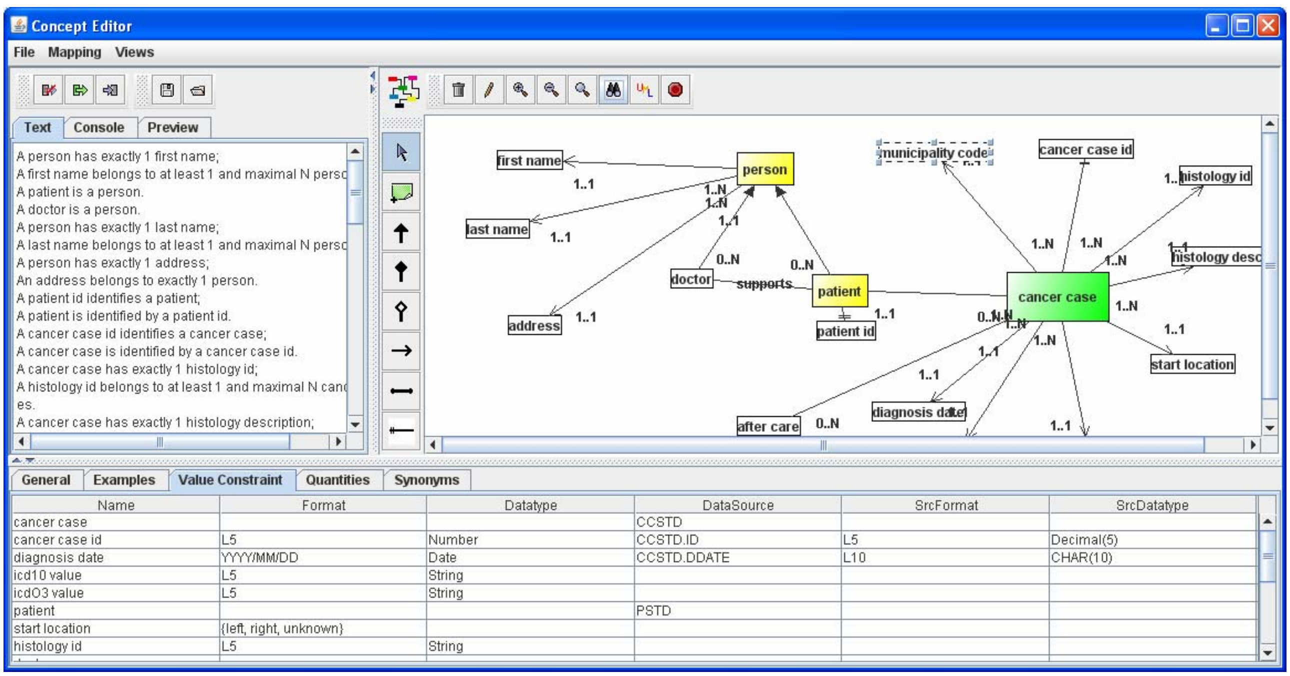


Figure 8: presentation of important concepts

A Data Quality Practical Approach

Maria del Pilar Angeles
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F
pilarang@unam.mx

Francisco García-Ugalde
Facultad de Ingeniería
Universidad Nacional Autónoma de México
México, D.F
fgarciau@servidor.unam.mx

Abstract **This paper describes a Data Quality Framework and its application within a Data Quality Project for heterogeneous multi-database environments. The quality assessment of derived data was performed by considering data provenance and conflict resolution functions. A Data Quality Assessment tool provides information regarding the elements of derived non-atomic data values. The assessment and ranking of non-atomic data is possible by the specification of quality properties and priorities from users at any level of experience. Consequently, users are able to make effective decisions by trusting data according to the description of the conflict resolution function that was utilized for fusing data along with the quality properties of data ancestors.**

Keywords- data quality; quality assessment; derived data; cleansing; data integration

I. Introduction

Multi-database systems provide integrated access to autonomous, distributed, and heterogeneous database systems. The process of data integration requires fusing conflicting data through the use of conflict resolution functions. Therefore, when users retrieve data from disparate data sources, they have no information about the corresponding components and how they were integrated.

This paper is based on previous work regarding the assessment of derived data by considering conflict resolution functions shown in [1], as part of a Data Quality Manager (DQM), which is a prototype to assess data quality and inform users about qualitative characteristics of integrated data, the elements it comes from and how it was fused in order to trust data according to its quality. The aim of this document is to propose a Data Quality Framework (DQF) within a heterogeneous multi-database context, and to present its implementation within a data quality project.

The Data Quality Manager implementation corresponds to the Data Quality Assessment element of the Data Quality Framework, but it could be part of any Data Quality Project life cycle. The DQM provides qualitative information that can be used to determine the current state of data, the business impact of erroneous data and the possible root causes of poor data quality.

We have already identified generic and usable quality criteria to measure and assess data quality of primary data sources, and integrated data at multiple levels of granularity in [2] and [3].

During the data integration process, data administrators require developing conflict resolution functions in order to solve data discrepancies. We enhanced the data lineage algorithm we developed in [4] to trace back the conflict resolution functions in order to provide further quality information to users.

The DQM implementation was based on a Framework for Data Quality Assessment developed in [2][3][4] composed by the identification of quality properties, its corresponding metrics, the process of assessment by data provenance, analysis of data quality, and ranking of data sources.

The implementation of our Data Quality Framework allowed users to determine causes of data quality problems and refine the data quality through data cleansing, monitoring, ensuring data quality during data production process, improvement, etc.

The outline of this paper is organized as follows. We briefly present a data quality overview in Section 2. Related work is described in Section 3. Section 4 describes a framework for conducting data quality projects. Section 5 explains the Data Quality Assessment Process as an element of the previous framework. Section 6 presents a practical approach by following the Data Quality Framework proposed. The last section concludes with relevant and novel features of the research and outlines future work.

II. Data quality overview

This section presents a generic overview of data quality, starting from commonly causes of data quality degradation, the impact of low information quality, the cost of data cleansing and our perspective for addressing data quality issues.

A. Data Quality Definitions

The subjective nature of the term Data Quality (DQ) has allowed the existence of general definitions such as “fitness for use” in [18], which implies that quality depends on customer requirements.

The definition established by Redman et al in [33], suggests that data quality can be obtained by comparing two data sources “A datum or collection of data X is of higher or (better) quality than a datum or collection of data Y if X meets customer needs better than Y”.

Another definition is “The distance between data views presented by an Information System and the same data in the Real World” in [34], which means that quality depends on the capacity of an information system to represent facts of the real world. Consequently, careful handling of data shall be done during its life cycle.

Recently, data quality has been defined as “the capability of data to be used effectively economically and rapidly to inform and evaluate decisions” [32].

However, these definitions are not very useful when data quality requires to be evaluated. Consequently, data quality rather than being defined has been characterized by multi attributes or dimensions according to specific application domains, types of assessment or customer requirements for instance, that shall be accomplished in order to be suitable for use.

As the determination of data quality is by comparing its corresponding attributes [30], [33], this collection of attributes must be defined, classified, measured and compared in order to determine an overall quality.

However, quality properties are often of a quantitative or qualitative nature, the former being easy to measure, but not the latter, which are subject to personal expertise.

Furthermore, “..What may be considered good quality information in one case (for a specific application or user) may not be sufficient in another case” [31], which means that even defining the quality attributes, and identifying their corresponding measures and assessment methods, the overall quality will depend on the specific priorities given by data consumers.

From our point of view, data quality is a multidisciplinary area, which involves management, statistics and computer sciences. We consider data quality not as the end but the means for making informed decisions.

The relevant data quality properties, its priorities, and the level of expected data quality depend not only on the data consumer experience, but also on the underlying type of information system.

B. Causes of Data Quality Degradation

Data are being deteriorated by processes bringing data from outside; incoming data may be incorrect and simply migrate from one place to another such as data conversion, batch feeds or real-time interfaces.

High volumes of data degradation are also introduced by wrong designed Extraction, Transformation and Loading (ETL) processes.

Data errors arise due to processes that manipulate the already existing data in the database such as periodic system updates with improper integrity constraints implementation.

Data are impacted by changes that for any reason are not captured, and wrong designed processes changing data from within.

There are some other processes that cause accurate data to become inaccurate because time related data changes over time and those changes are not reflected in the system.

C. Impact of low Data Quality

Poor data quality might affect every sector of industry such as finance [24], where an error attributed to the New York Stock Exchange resulted in several inaccurate stock quotes being picked up and posted at a number of news and investment organizations; within the medicine sector [25] a woman underwent a double mastectomy after being advised that she had breast cancer. After the surgery she was informed that the laboratory had switched her lab results with another patient and that she never had cancer; in the Academy sector [26], a University emailed 1,700 applicants to announce their acceptance into the class of 2007.

Unfortunately, 550 of the applicants received the letter in error they had already received rejection notices. The error was attributed to a "systems coding error". However, there is a possibility that the acceptance status of the 550 students was updated by mistake after sending the rejection notice.

Users should be aware of the quality of data they are accessing along with the cause of its degradation. For instance, identifying which data are time-related becoming obsolete as time goes by; quality of data might be application-related due to missing or wrong designed constraints; integrated data have been passing from one application to other or from one data source to other through data fusion or transformation; etc.

D. The cost of data cleansing

According to T. Anderson in [22] the cost of poor data quality is the sum of the cost to prevent errors and the cost to correct them and the cost to make them good for the customer. Pragmatically speaking, the cost of poor data quality extends far beyond the cost to fix it.

The Data Warehousing Institute estimates that data quality problems currently cost U.S. businesses over $600 billion annually. Errors are very hard to repair, especially when systems extend far across the enterprise, and the final impact is very unpredictable.

The first reaction at cleansing personal details would be determining if a single record is "correct" by calling the corresponding telephone number, and ask the person whose name shares the record with the telephone number. If the person comes to the phone, ask if all the values are accurate, and correct those that are not. If there is no one there by that name, the record is incorrect. The next step in data cleansing requires additional information, and if none is available, then the algorithm ends. This is a simple and accurate algorithm. However, commonly is neither cost effective nor scalable because depends on the number of records, staff members and telephones. Automated solutions may be more scalable, more costly, less accurate, more complex, require more expertise, etc.

D. Loshin in [23] states that the cost of cleansing data requires to analyze which is the size of data in number of records and columns, which would be the criteria in order to define data “clean”, if the relevant data are in a single table or scattered across many data sources, and the number and level of experience of customers. The level of reasonable effort for spending on data cleansing must be less than the value of the accrued business benefits, and this provides an upper limit to what could be budgeted for the process.

The subject of this work is concerned with the specification and implementation of a Data Quality Framework for the identification, measurement, and assessment of data quality of derived data, and data sources at any level of granularity to provide ranking of data sources based on the user specified context. After the data quality diagnosis, feasible data cleansing within a monitoring process shall be possible, according to the business requirements and the level of data quality pre-established.

As low data quality impacts on business, and the process of assessing and cleansing data is not trivial, important research has been done recently. Section 3 presents recent developed frameworks for data quality projects, how previous approaches have dealt with data inconsistencies during data integration and how the assessment of data quality has been addressed in particular.

III. RELATED WORK

A. Data Quality Frameworks

The Massachusetts Institute of Technology (MIT) and the Cambridge Research Group, among other institutions, have co-founded the MIT Total Data Quality Management program (TDQM) [28]. The aim of TDQM is to create a theory of data quality based on disciplines such as Computer Science, Statistics, and the Total Quality Management field, and is focused on the definition and measurement of data quality, the identification and analysis of data quality impact, and the redesign of business practices and implementation of new technologies to improve information quality.

In Total Data Quality Management the concepts, principles and procedures are presented as a methodology, which defines the following continuous life cycle: define, measure, analyze and improve data as essential activities to ensure high quality, managing information as a product.

There are more detailed approaches such as the one proposed by D. McGilvray in [19] who proposes ten steps for executing data quality projects. The main objective of data quality projects is to achieve a reasonable level of quality that brings success to companies. Therefore, the project starts by the identification of business needs. After an analysis of information environment it is possible to identify the essential data and information corresponding to those business needs. During the assessment of data quality as a third step, the design and implementation of an assessment plan for relevant data is a key in order to evaluate the current state of data. As the following step, the assessment results should be analyzed and documented to determine the business impact of poor quality of relevant data. Step 5 corresponds to the identification of root causes of data issues and initial recommendations. The sixth step is the development of improvement plans. The implementation of the improvement plan will correct current data errors, and prevent future data errors (steps 7 and 8). Step 9 is concerned with monitoring if the improvement plan is providing the expected results through implementing controls allows finishing the cycle and starts it over again. However, communicating actions and results along the whole process is a key for success.

David Loshin in [23] identifies 17 steps required for data quality management.
The first step is to recognize the problem, if there are some issues that are affecting the business then there is evidence that poor data quality is having an impact in order to determine whether such evidence points to any particular problems with data quality or not.
The second step is to obtain the management support by showing them how the business is affected or can be affected by poor data quality, and at the same time their support and enforcement of a data ownership policy document for guiding the roles associated with information and the responsibilities accorded those roles. The third step is to spread the word by a data quality education program. The forth step is mapping the information chain in order to understand how information flows through the organization, which is a chart that describes processing stages and the channels of communication between them. Data Quality Scorecard is the fifth step, which is concerned with the overall cost associated with low data quality and can be used as a tool to help determine where the best opportunities are for improvement. The sixth step is to perform a current state assessment to obtain information regarding the causes of data quality issues, this step requires identifying which data quality dimensions will be relevant and identifying points within the information chain and for measuring for understanding the scope and magnitude of data quality problems. The seventh step is requirements assessment, which is in charge of problems prioritization, assigning responsibility and creating data quality requirements for identifying the location in the information chain where the requirement is applied, a description of the measurement rule, the minimum threshold for acceptance among others. Step eight is choosing the first problem to address. Therefore such problem should have a noticeable impact in order to ensure the continued operation of the data quality program. The next step is regarding to build the team to solve the problem. The step ten is related to the identification of proper data quality tools in order to support data cleansing, data standardization, etc. The eleventh step is to define a metadata model to store enterprise reference data. The next step is the definition of data quality rules. Step 13 is related to the Archaeology/Data mining to look for data domains, mappings, and data quality rules that are embedded in data. The fourteenth step is for managing suppliers, a corresponding program will be required to impose requirements on external data suppliers to specify the rules that are being asserted about expectation of the data along with penalties for nonconformance. Step fifteen is concerned with actually executing the data improvement. The next step is related to measuring the improvement in order to demonstrate success at improving data quality by performing the same measurements from current state assessment. The last step is to build on each success. Each small success

should be used as leverage with the senior level sponsors to gain access to bigger and better problems.

For the above mentioned frameworks we can say that there is no consideration of data quality within heterogeneous multi-database environments or enterprise information integration contexts, where data come from a number of data sources facing semantic and syntactic heterogeneities and derived data are product of integration processes.

B. Previous Approaches of Data Quality Assessment

A particularly important element within data quality projects is the data quality assessment. Therefore, this section presents previous approaches of data quality assessment.

Gertz developed some data integration techniques in [9], based on data quality aspects within an object oriented data model, and data quality information stored in a metadata. In the case of data conflicts between semantically equivalent objects, the object with the best data quality must be chosen. However, the quality goals specification limits the possibility of more combinations of priorities from the user, because they are not given in weights or percentages, just the “the most accurate” or “the most up to date”. Consequently, not just one or two combinations of quality priorities will satisfy users. One result might be good enough for one user under a specific situation, but of poor quality for other.

The project Multiplex directed by Motro and Rakov [11] was based on accuracy and completeness as quality criteria. A voting scheme, using probabilistic arguments, identifies the best set of records to provide a set of ranked tuples to the user, but no further information about their associated quality. Therefore, users are neither able to establish their quality preferences or priorities nor to take part in the resolution process.

The project Quality-driven Integration of Heterogeneous Information Systems was developed by F. Naumann in [12]. The aim was to identify and to rank high quality plans, which produce high quality results. There is a classification of specific quality criteria according to the level of granularity (in this approach data sources, queries and attributes). However, there is no further specification of how to assess quality at different levels of granularity. Data sources are ranked using the DEA method. Therefore, there is no consideration of user priorities for this process. Besides, subjective criteria are used for discarding data sources such as reputation and understandability.

The aim of the Data Quality in Cooperative Information Systems (DaQuinCIS) project [15] was to define an integrated framework to improve data quality in cooperative environments. Such a framework started from the Total Data Quality Management methodology which was extended to suit the cooperative information systems requirements, and supporting data quality monitoring and improvement. The use of a metadata was required to store the quality score, the meaning of the quality value, and how the measurements were carried out. This approach takes into account the specification of data granularity as the combination of elementary data items that are subject to quality metrics. There is also a difference between computing the quality of aggregated data and computing an aggregate indicator over a set of items. However, the measurement is not only subjective but also different methods are utilized to measure quality, yielding different results. Furthermore, data derived from multiple data sources is not considered.

A Generic Framework of Information Quality was developed by Burgess in [8] with around 60 information quality properties classified hierarchically according to time, utility and cost. Nevertheless, this approach was focused on information search not on measurement and assessment of quality at data value level.

A. Maydanchik proposes a methodology in [10] for data quality assessment to identify all data errors. In order to do so the project shall involve business users, IT specialists, data quality experts to a project team.

The data quality project plan which in turn consists of four steps a) planning for identifying project scope and objectives; b) preparation for gathering relevant data and metadata; c) implementation concerned with designing the data quality rules, and d) fine tuning, where data experts validate error reports in order to enhanced data quality rules.

It is desirable to monitor data quality on an ongoing basis, in order to see data quality trends, identify new data problems, and check the progress of data quality improvements initiatives.

Within the implementation phase of the data quality assessment, data quality rules can be executed automatically in order to find such data errors, the first step is design, cataloguing, and coding data quality rules. The second step for data quality assessment is the process to identify and eliminate rule imperfections by manual verification of the sample data by data experts, the analysis of sample verification findings and the search for patterns; and to enhance the rules to eliminate as many flaws as possible; and repeat until obtain the expected results. The third step is concerned with storing information about all the identified data errors in an error catalogue in order to identify and analyze error patterns and enhance data quality rules and identify how to correct data errors. The next step is to identify and tabulate aggregate data quality scores. Accurate data quality scores help to translate data quality assessment results into cost of bad data, return of investment from data quality improvement and expectations from the projects. The fifth step is to identify the content and functionality of the data quality metadata warehouse which contains tools for organization and analysis of all meta data relevant to or produced by the data quality initiatives, contains aggregate meta data, rule metadata atomic metadata and general meta data. The last step is the recurrent data quality assessment for an ongoing data quality monitoring.

When data quality assessment is done on a regular basis and if the target database contains large volumes of data, running the rules directly against the production database might be a better solution than replicating it to the staging are data quality assessment is technically and technologically challenging, the best solution depends on the dynamics of the data.

C. Important Remarks

Within the previous approaches, there is no consideration of derived data. Data in all of these approaches has been considered as a product of a primary source. However, due to the explosion of information over the last decade, we cannot assume that any data source is necessarily the point of origin of the data users require. Hence, the fundamental presumption of current data management practice, the "Presumption of Primary Authorship" must be challenged.

Users should be provided with information regarding data as an atomic value, or if it is composed data, what the atomic values were and the quality generated from. This challenges the "Presumption of Atomicity".

The assessment methodologies presented until now do not consider data provenance as part of root causes of poor data quality. Cleansing derived data with no consideration of data fusion or conflict resolution functions is not an effective solution for assessing data within heterogeneous multi-database environments.

The next three sections present our Data Quality Framework, a Data Quality Manager prototype as an implementation of the Assessment of data quality and a practical application of both of them.

IV. THE DATA QUALITY FRAMEWORK

We propose a framework for Data Quality composed by seven phases. The first phase is the identification of data quality problems by their impact on the business, considering data quality experts, data domain experts and end users of any level of experience. The second phase is the identification of relevant data that has direct impact on the business for an estimation of poor data quality cost. The third phase is the creation, identification, or modification of relevant business rules. Commonly, some business rules have not been considered during the application development or they might exist but require enhancement. The fourth phase is the Data Quality Assessment Model for the analysis of data quality at different levels of granularity considering data provenance. The analysis of data quality assessment enables expert users to establish different priorities to quality properties and different levels of granularity for assessment. The fifth phase corresponds to the determination of the business impact through data quality comparison. The difference between the expected data quality and the actual data quality scores will establish the feasibility of the data quality project for cleansing and continuous assessment and the business impact in terms of operational efficiency, or increased revenue, money saved, etc. The sixth phase corresponds to the cleansing of data by enforcing the business rules, data standardization, and data matching. The last phase corresponds to monitoring data quality and executing the assessment phase on regular basis.

The proposed Data Quality Framework is simple enough to be suitable to any size of data quality project, and at the same time its data quality assessment element considers data provenance, data fusion and conflict resolution functions for comparing and resolving extensional inconsistencies within virtual or materialized data integration.

Fig. 1 shows the elements of the Data Quality Framework.

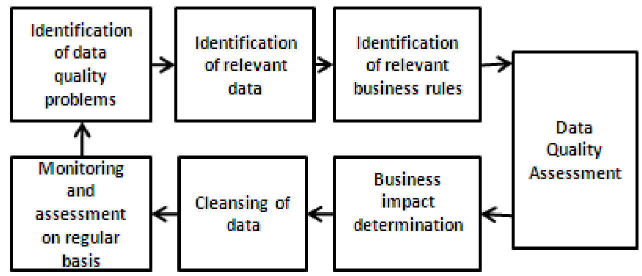


Figure. 1 The Data Quality Framework

In Section 5 we explain in more detail the Data Quality Assessment element of the Data Quality Framework.

V. THE DATA QUALITY ASSESSMENT PROCESS AND ITS IMPLEMENTATION

A. The Data Quality Assessment Process

The first step corresponds to the identification of useful data quality properties for the measurement, and assessment of data quality of derived data, and data sources at multiple levels of granularity, to provide data consumers with qualitative information directly related to the relevant data and business rules identified during the first three steps of the Data Quality Framework. The outcome of this step is called a Data Quality Reference Model, which contains an objective and effectively set of quality criteria to provide an unbiased measure of quality to users at any level of experience they might have. A generic set of data quality properties has been classified and summarized according to different user perspectives such as internal and external focuses or representation, value, and context in [3].

As we are addressing any level of experience user, the aim of the second step is to discuss which existing metrics are suitable for an unbiased, and user independent estimation of data quality scores to provide a more objective quality metadata. The development of new metrics is not relevant for this research, but to extend existing metrics to assess data quality at different levels of granularity. Therefore, the outcome of this step is called a Measurement Model [4], which assembles and extends the already existing data quality metrics [6] [11] [14] for the measurement at database, relation, tuple, and attribute levels of granularity.

The third step is concerned with the identification of methods required to represent, to interpret, and to assess data quality indicators. The assessment methods utilised should provide meaningful and useful scores. Therefore, objective criteria, and process criteria should be included in the Assessment Model which are user independent, rather than subjective criteria, which can only be determined by individual users based on their experience and background.

The Assessment Model provides a mechanism for tracking data lineage for the assessment of quality of derived data. Previous approaches work from the presumption of primary authorship and the presumption of atomicity. Therefore, the utilization of data lineage as a mechanism for assessing data sources at different levels of granularity, challenging the presumptions of primary authorship and atomicity are novel.

The forth step corresponds to the estimation of the quality scores of primary data sources, which will be stored in a Quality Metadata.

The fifth step is the assessment of derived data, which requires the definition and population of a provenance metadata. The assessment is based on the quality scores of their corresponding ancestors, and the computed scores are also stored in a Quality Metadata.

The sixth step presents two options for the analysis of data quality, according to user requirements and business information stored in the organizational metadata

a) The selection of the best data sources before the query execution on the bases of its quality scores. Therefore, the consideration of data quality scores helps the query planning by finding the best combination of data sources for the execution plan.

b) The comparison of data quality aggregated scores corresponding to different query plans for the same business question.

The seventh step is the ranking of data sources, where the data quality scores previously stored in the metadata are used as a whole with their corresponding priorities stated by the user. Fig. 2 shows the Data Quality Assessment Process.

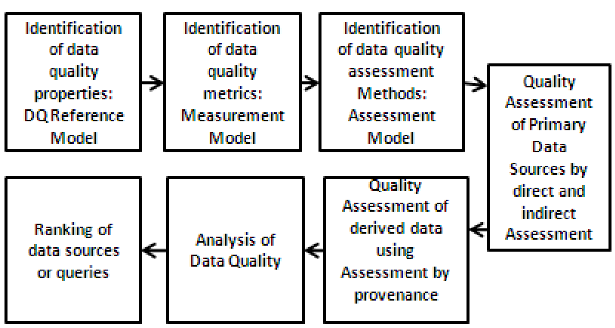


Figure. 2 The Data Quality Assessment

B. *The Data Quality Manager*

The process of assessment of data quality has been developed within the Data Quality Manager through the implementation of the already mentioned models and a quality metadata, a provenance metadata, and an organizational metadata.

The Quality Metadata is a repository to contain the quality scores per each data source obtained during the data quality assessment process, and reloaded to assess at lower levels of granularity.

The provenance metadata is a repository to contain ancestors' information for the tracking of provenance of the participant data sources.

The Organizational Metadata is a repository to contain the information required to map from the global schema to the local schema in order to resolve intensional inconsistencies (semantic differences) within the multidatabase environment, for further information regarding intensional inconsistencies, please refer to [29]. The organizational metadata will also contain business rules and relevant information for business understanding.

The DQM is part of a diagnostic pre-process for data cleansing, or after data cleansing to evaluate data quality improvement.

The DQM represents the data quality assessment component of the Data Quality Framework. The DQM is designed to utilise data quality measures to provide qualitative information. As we have explained, such information could be further used within the data integration processes.

The Data Quality Manager (DQM) is a system designed specifically for centralized processing of multiple interfaces between multiple databases; it allows maintaining detailed data provenance and data quality metadata for future reference.

The architecture of the DQM is shown in Fig. 3.

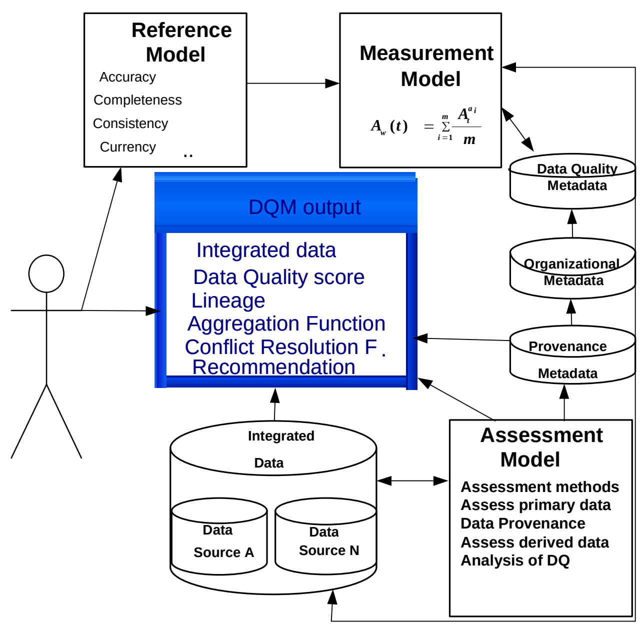


Figure. 3 Components and outcomes of the Data Quality Manager

The DQM provides qualitative information to any level of experience users to extend the scope and range of information available relative to the integrated data within the quality properties and priorities they state.

The DQM in the case of naive users provides an appropriate combination of scaling with ranking methods. In the case of expert users, they will have the ability to define scaling, ranking, quality properties and the priorities for a higher level of analysis. Users should be able to select the quality priorities. The specification of Multi-attribute

Decision and Scaling Criteria methods is also possible by experienced users.

The functionality and capability of the Data Quality Manager prototype has been validated against the specifications based on a testing plan detailed in [5]. We have also demonstrated that the DQM provides appropriate scores according with the expected outcomes based on the actual quality of data and information relative to the conflict resolution function utilized during the integration process.

VI. A PRACTICAL APPROACH FOR DATA QUALITY

This section is aimed to explain the implementation of our Data Quality Framework within a data quality project, and is intentionally more focused on the results presented by the Data Quality Manager for the assessment of derived data.

As the Data Quality Manager (DQM) tool is aimed to work within a multi-database environment, the conducted tests are based on a TPC Benchmark™H (TPC-H) [17].

TPC-H is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance. This benchmark illustrates decision support systems that examine large volumes of data, execute queries with a high degree of complexity, and give answers to critical business questions. The names of the implemented databases are TPCH, TPCHA and TPCHB.

A. Data Quality Issues

Users are unable to make informed decisions because they are retrieving different results for the same query. The problem is also called extensional inconsistencies, and it refers to the data value differences between the participating data sources during data integration. The cause of extensional inconsistencies is that queries can be executed on different data sources semantically equal. For further information regarding extensional inconsistencies, please refer to [29].

In order to determine business needs we require a list of the most important business queries, after the identification of such queries, executive users prioritize the queries according to their impact on business. Focusing in what is relevant and appropriate is critical for finding relevant data.

At this point the analyzed data, processes, technology, and people allows a better understand of all these components and their impact on information quality.

B. Relevant Data

The identification of relevant data affecting business questions was performed by the identification of such conflictive business queries. This paper will present just three queries corresponding to one possible option. However, similar analysis shall be done for each semantically equal business question executed on different data sources. The important business questions identified are Customer Distribution, Product Type Profit Measure, and National Market Share.

The business query called Customer Distribution seeks relationships between customers and the size of their orders. It determines the distribution of customers by the number of orders they have made, including customers who have no record of orders, past or present. It counts and reports how many customers have no orders, how many have 1, 2, 3, etc.

A check is made to ensure that the orders counted do not fall into one of several special categories of orders. Special categories are identified in the order comment column by looking for a particular pattern. Please refer to [17] for further detail. The query `Cus_Distribution` has been integrated by the outer join of two tables `CUSTOMER` and `ORDERS`, and the relevant data columns are `C_CUSTKEY`, `O_ORDERKEY` and `O_COMMENT`. The SQL Text of the `Cus_Distribution` query is presented as follows.

```
SELECT C_CUSTKEY AS C_COUNT,
       COUNT (O_ORDERKEY) AS HOW_MANY
FROM
CUSTOMER LEFT OUTER JOIN ORDERS ON
          C_CUSTKEY = O_CUSTKEY
         AND O_COMMENT NOT LIKE
          '%UNUSUAL%DEPOSITS%'
GROUP BY C_CUSTKEY
```

The Product Type Profit Measure business question finds for each nation and each year, the profit for all parts ordered in that year which contain a specific substring in their part names and which were filled by the Supplier in that nation. The corresponding instantiation of the business question is called `pt_profit` and it contains relevant data such as `PART.P_PARTKEY`, `PART.P_NAME`, `SUPPLIER.S_SUPKEY`, `LINEITEM.L_SUPPKEY`, `L_PARTKEY`, `L_ORDERKEY`, `PARTSUPP.ORDERS` and `NATION.NATIONKEY`.
The SQL text code of the query `pt_profit` is presented below.

```
SELECT      N_NAME AS NATION,
 EXTRACT(YEAR FROM O_ORDERDATE) AS YEAR,
 L_EXTENDEDPRICE * (1 - L_DISCOUNT) -
PS_SUPPLYCOST * L_QUANTITY AS AMOUNT
FROM PART, SUPPLIER, LINEITEM, PARTSUPP,
ORDERS, NATION
WHERE S_SUPPKEY = L_SUPPKEY
AND PS_SUPPKEY = L_SUPPKEY
AND PS_PARTKEY = L_PARTKEY
AND P_PARTKEY = L_PARTKEY
AND O_ORDERKEY = L_ORDERKEY
AND S_NATIONKEY = N_NATIONKEY
AND P_NAME LIKE '%MINT%'
```

The National Market Share business question shows the market share for a given Nation within a given Region. It is defined as the fraction of the revenue from the products of a specified type in that Region that was supplied by Suppliers from the given Nation. The query determines this for two years. The relevant data are PART.P_PARTKEY, PART.P_TYPE, SUPPLIER.S_SUPPKEY, LINEITEM.L_PARTKEY, LINEITEM.L_SUPKEY, ORDERS.O_ORDERKEY, ORDERS.O_ORDERDATE, ORDERS.O_CUSTKEY, CUSTOMER. CUSTKEY, NATION.N_NATIONKEY AND REGION.R_NAME. The SQL text code for the corresponding query C_Market_Share is shown as follows.

```
SELECT      EXTRACT(YEAR FROM
O_ORDERDATE) AS O_YEAR,L_EXTENDEDPRICE *
(1 - L_DISCOUNT) AS VOLUME,
      N2.N_NAME AS NATION
FROM  PART,SUPPLIER,    LINEITEM,ORDERS,
      CUSTOMER,NATION N1,NATION N2,
      REGION
WHERE P_PARTKEY = L_PARTKEY
      AND S_SUPPKEY = L_SUPPKEY
      AND L_ORDERKEY = O_ORDERKEY
      AND O_CUSTKEY = C_CUSTKEY
AND C_NATIONKEY = N1.N_NATIONKEY
AND N1.N_REGIONKEY = R_REGIONKEY
AND R_NAME = 'AMERICA'
AND S_NATIONKEY = N2.N_NATIONKEY
AND O_ORDERDATE BETWEEN DATE 'date' AND
DATE 'date'
AND P_TYPE = 'LARGE PLATED NICKEL'
```

C. Business Rules

Once obtained the relevant data, the next step is to identify their corresponding business rules. They shall be enforced within the relevant data in order to detect data errors and correct them.

In the case of the business questions National Market Share and Product Type Profit, the corresponding trigger that inserts a new tuple into REGION whenever a new tuple is inserted into NATION, and the trigger that inserts a new tuple into NATION whenever a new tuple is inserted into REGION were enforced.

D. Assessment of Data Quality

Data quality assessment tells us about existing data problems and their impact on various business processes. When done recurrently, it also shows data quality trends.

The elements of the Data Quality Assessment Process produced during the practical approach will be explained in detail in the following subsections.

Data Quality Properties

Considering the relevant data and business rules, the identification of which quality properties are relevant for assessment is required. However, according with Lee and Strong in [21], the responses from data collectors, data custodian, and data consumers within the data production process determine data quality because of their knowledge.

Data collectors are associated to the quality properties accuracy, accessibility, relevance, completeness and timeliness. Data Consumers are more interested in the accuracy of and uniqueness of data in order to use them for making decisions. Their research was oriented to determine the causes of poor data quality during the data life cycle and how the knowledge of the participant users reflects the quality of data. Therefore, the identification of the relevant quality properties is also directly related to the knowledge of the data according to the experience of users.

In this Data Warehouse context, the quality criteria vary depending on the data source, for example for look up tables there will be low volatility, but accessibility is important. In case of Fact tables, as they provide the sales detail, accuracy, uniqueness, and completeness are important because they would be directly reflected in the generation of aggregate data in the summarize tables.

The integration of data sources that contain duplicated tuples could result in extensional inconsistencies. Therefore, the quality property called uniqueness should be included as a relevant quality criterion for the assessment of data quality to help in the resolution of extensional inconsistencies.

A Generic Data Quality Reference Model has been discussed in [2]; it is suitable to any application domain and supports the full range from the internal focus to the external focus.

After an analysis of the proper quality properties according to the expert users, the type of information system and the relevant data identified, we have reduced the number of quality properties from the Generic Reference Model to those corresponding to the data value level in order to obtain results fast for a rapid return on investment (ROI). Therefore, the quality properties or data quality dimensions used for this assessment are accuracy, completeness, consistency, currency, timeliness, uniqueness and volatility.

Data Quality Metrics

Designing the right metrics is the most challenging task during the process of data quality assessment. However, the challenge is to design them and make sure that they indeed identify all or most errors, avoiding metrics that reflect the same error in many different ways and produce comprehensive error reports.

Once identified the relevant quality properties the next step is to assess them through the measurement model, and synthesize the results from the assessments.

The Measurement Model corresponds to the metrics for data quality properties identified in previous step, and to the business rules already identified.

Accuracy is the measure of the degree of agreement between a data value or collection of data values and a source agreed to be correct. [27].

Completeness is the extent to which data is not missing [14] and is divided by two quality dimensions: coverage and density in [12].

Consistency is the extent to which the values are the same for overlapping entities and attributes. Data are consistent with respect to a set of constraints if they satisfy all constraints in the set [11]. Often referred as integrity constraints state the proper relationships among different data elements" [14]

The following SQL text code shows the measurement of referential integrity between `LINEITEM` and `PART and LINEITEM` and `SUPPLIER` as one of the requirements for the query `C_Market_Share`. Finally, the data quality score is stored in the quality metadata through an insert-select sentence.

```
/* lineitem with part */
begin
declare @part real
declare @supplier real
select @part=
case
when convert(real,count(L_PARTKEY))=0
then 1
when  convert(real,count(L_PARTKEY))> 0
then convert(real,count(L_PARTKEY))
end
from lineitem
where not exists (select * from part
                  where
P_PARTKEY=TPCHA.dbo.lineitem.L_PARTKEY)
/* lineitem with supplier*/
select @supplier=
case
when convert(real,count(L_SUPPKEY))= 0
then 1
when  convert(real,count(L_SUPPKEY))> 0
then convert(real,count(L_SUPPKEY))
end
from lineitem
where not exists (select * from supplier
                  where
S_SUPPKEY=TPCHA.dbo.lineitem.L_SUPPKEY)
select
object_id,12,@part,@supplier,mrows,"1-
inconsistent/total rows"
from Metadata.dbo.numrows
where object="TPCHA.dbo.lineitem"
group by object_id,mrows
insert Metadata.dbo.Scores
select object_id,12,1-
((@part/convert(real,mrows))*(@supplier/
convert(real,mrows))),"1-
inconsistent/total rows"
from Metadata.dbo.numrows
where object="TPCHA.dbo.lineitem"
group by object_id,mrows
end
```

Currency is the time interval between the latest update of a data value and the time it is used [11].

Timeliness is the extent to which the age of data is appropriate for the task at hand [6], and is computed in terms of currency and volatility. Timeliness has also been presented as context related dimension.

Uniqueness is the extent to where an entity from the real world is represented once. The below SQL code computes the ratio between the number of non-unique rows and the total number of rows in the nation relation.

```
insert into Scores select 301,2,
convert(real,count(distinct
N_NATIONKEY))/convert(real,count(*))
,"non-duplicated/total values"
from TPCHA.dbo.nation
```

Volatility is the interval of time where data remains valid on the system and is related to the update frequency [6].

Assessment Methods

Most metrics proposed until now are just at one level of granularity. Particularly, completeness has been deeply approached in [12] and [20] with the coverage and density concepts in the former, and at different levels of granularity in the latter. However, we have taken into account not only attribute, and relation levels of granularity following the completeness example given in [20] but also the database level. We are considering the cardinality of a relation when measuring its quality. Therefore, the estimation of quality at database level is taken from the average score of its relations as a representative aggregation function.

The strictness of data quality assessment is a weak or strong characterization depending on evaluating the quality property as a percentage or as a Boolean function respectively [20]. The strong characterization of the quality metrics is useful in applications in which it is not possible to admit errors at the corresponding level of granularity. For instance, in the case of accuracy at tuple level, it would be useful if and only if all the instances of its attributes are accurate. The remainder of this section presents 16 formulas corresponding to the relevant quality properties already identified, for further information regarding such formulas please refer to [5].

In this practical approach the assessment of data quality considers the weak strictness to make possible the comparison of data sources for a number of data quality properties. However, as there might be alternatives where strictness could depend on the level of quality required, according to specific applications we present both characterizations.

During the assessment of data quality, we identified a granularity-based assessment classification according to the level of granularity in which the quality assessment is done a) Direct assessment; b) Indirect Assessment; and Assessment by provenance.

Assessment of primary data sources

Direct assessment is the process of assessment that relates directly to the level of granularity. For instance, the uniqueness dimension U(t_j), which relates at the tuple level.

$$U(t_j) = 1 \quad if \quad tuple \quad j \quad is \quad represented \quad once \quad in \quad a \quad relation$$
$$U(t_j) = 0 \quad otherwise \ldots\ldots (1)$$

Accuracy at value level corresponds to the presence of the correct data value within a specific attribute a_i in a tuple t, and is set by the following notation:

$$A_t^{a_i} = 1 \quad if \quad value \quad in \quad a_i \quad is \quad correct$$
$$A_t^{a_i} = 0 \quad otherwise \ldots\ldots (2)$$

b) Indirect assessment: The score is calculated based on other scores at other levels of granularity of the same source. For example,

Weak accuracy at attribute level $A_w(a_i)$ is the number of tuples with correct values for a specific attribute a_i divided by the cardinality of the relation S.

$$A_w(a_i) = \sum_{j=1}^{n} \frac{A_{t_j}^{a_i}}{n} \ldots\ldots (3)$$

The accuracy of an attribute $A_s(a_i)$ is strong if all instances t_j of the attribute a_i in the relation S are correct.

$$A_s(a_i) = 1 \quad if \quad A_{t_j}^{a_i} = 1 \quad \forall j \in [1..n]$$
$$A_s(a_i) = 0 \quad otherwise \ldots\ldots (4)$$

Weak relation accuracy $A_w(S)$ is the number of tuples where every attribute is correct divided by the total number of rows.

$$A_w(S) = \sum_{j=1}^{n} \frac{A_s(t_j)}{n} \ldots\ldots (5)$$

Strong relation accuracy $A_s(S)$ is that when all the tuples contain correct values in every attribute, or when a relation contains strong tuple accuracy, and strong attribute accuracy.

$$A_s(S) = 1 \quad if \quad A_s(t_j) = 1 \quad , \quad \forall j \in [1..n]$$
$$A_s(S) = 0 \quad otherwise \ldots\ldots (6)$$

Then accuracy at database level $A(D)$ can be derived from the average of all accuracy scores at relation level.

$$A(D) = \frac{\sum_{k=1}^{w} A(S_k)}{w} \ldots\ldots (7)$$

Consistency at the relation level depends on consistency at the row level. The weak consistency at the relation level $Cn_w(S)$ is the percentage of tuples t_j with all instances of the attributes consistent.

$$Cn_w(S) = \sum_{j=1}^{n} Cn_s(t_j) \Big/ n \ldots\ldots (8)$$

Direct and indirect assessments are performed on the ancestors' data sources.

In the case of the data quality assessment cannot be computed directly for performance issues then if it is possible, the assessment by provenance is applied.

The following subsection is concerned with the quality estimation of integrated data as part of the Assessment of Data Quality.

Assessment of derived data

Assessment by provenance is the process of assessment when the score of an object is computed based on the quality indicators of its ancestors.

In order to explain how quality of derived data might be assessed through data provenance, consider a query or a source *s* that comes from *n* ancestors α_j.

For instance, accuracy of derived data $A(s)$ is computed by the average of the scores of its ancestors.

$$A(s) = \frac{\sum_{j=1}^{n} A(\alpha_j)}{n} \ldots\ldots (9)$$

Completeness of derived data $C(s)$ is determined by the average value of the completeness of its ancestors.

$$C(s) = \frac{\sum_{j=1}^{n} C(\alpha_j)}{n} \ldots\ldots (10)$$

Consistency of derived data $Cn(s)$ is determined by the average of the consistency of its ancestors. The consistency of its foreign keys is checked with its corresponding primary keys in each ancestor.

$$Cn(s) = \frac{\sum_{j=1}^{n} Cn(\alpha_j)}{n} \ldots\ldots (11)$$

The currency of derived data $Cu(s)$ is the greatest value of the corresponding currency measures from the different ancestors.

$$Cu(s) = \max\left(Cu(\alpha_j)\right) \quad , \quad \forall j \quad \in [1 \cdots n] \quad \text{......... (12)}$$

Volatility is the update frequency. When there are a number of data sources with different volatilities, the volatility of derived data $Vo(s)$ is the greatest value of the corresponding volatility measure from its different ancestors.

$$Vo(s) = \max\left(Vo(\alpha_j)\right) \quad , \quad \forall j \quad \in [1 \cdots n] \quad \text{........ (13)}$$

The following SQL code shows the implementation of the measurement of volatility considering the maximum volatility value from its ancestors.

```
 insert Scores
select 1210,8,max(score),"max(volatility
of ancestors)"
from Scores
where object_id in
 (select ancestor_id
  from Ancestors
  where object_id = 1210)
and criterionID=8
```

Uniqueness of derived data $U(s)$ is obtained from the average of its ancestor's uniqueness.

$$U(s) = \frac{\sum_{j=1}^{n} U(\alpha_j)}{n} \quad \text{..(14)}$$

Timeliness of derived data $T(s)$ is estimated in terms of its maximum currency and volatility.

$$T(s) = \max\left(0{,}1 - \frac{Cu(\alpha_j)}{Vo(\alpha_j)}\right) \quad , \quad \forall j \quad \in [1 \cdots n] \quad \text{......... (15)}$$

Consistency of derived data is determined by the average of the consistency of its ancestors. The consistency of its foreign keys is checked with its corresponding primary keys in each ancestor.

$$Cn(s) = \sum_{j=1}^{n} Cn(\alpha_j) \Big/ n \quad \text{.........................(16)}$$

During the assessment of data quality, the Data Quality Manager tool obtains information about the quality of the ancestors from which derived data was produced. Assessing the quality of the available primary data sources from which the integrated data has been obtained is addressed in case there is no possibility of computing data quality from the data itself.

Once obtained the quality properties of the ancestors, the Data Quality Manager is able to assign quality scores to derived data by the aggregation of the quality properties of its ancestors. This assessment requires that all the quality scores of the corresponding ancestors are available. A quality aggregation function combines components of quality into an overall quality specification.

The DQM can show quality scores of the ancestors or derived data by selecting them from the provenance tree, and a brief formula is shown in the Unit column in order to provide the metric from which it was computed.

Fig. 4 shows qualitative information based on data provenance of a query `Cus_Distribution`.

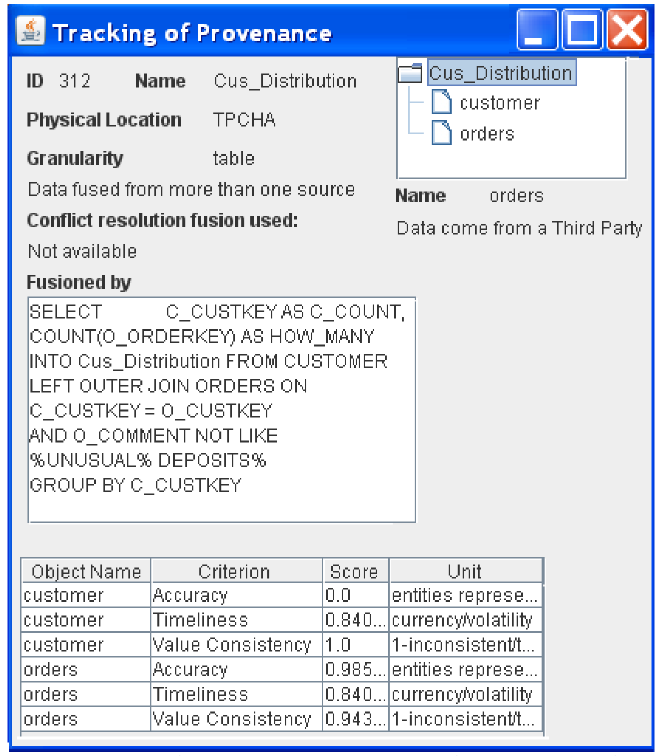


Figure. 4 Assessment of Cus_Distribution data quality from the quality of customer and orders, its ancestors

A statistically sound aggregation is when the quality property was obtained by mean values with given sample size n and one of standard deviation or standard error. If statistically soundness is to be preserved, a mean value can only be calculated for numeric values with an underlying normal distribution.

We have considered average as a default conservative aggregation function for accuracy, completeness, consistency, and uniqueness and a default pessimistic aggregation function for time related quality properties.

There might be different criteria for the aggregation of the qualitative measures. However, the DQM is able to ask expert users which aggregation function would they like to apply for the quality estimation.

Fig. 5 shows the quality estimation for `Cus_Distribution` given by the average of the scores from the ancestors in the case of accuracy, the maximum value as pessimistic approach for the assessment of timeliness.

Users are able to obtain their quality scores in order to decide whether `Cus_Distribution` is suitable for use or not.

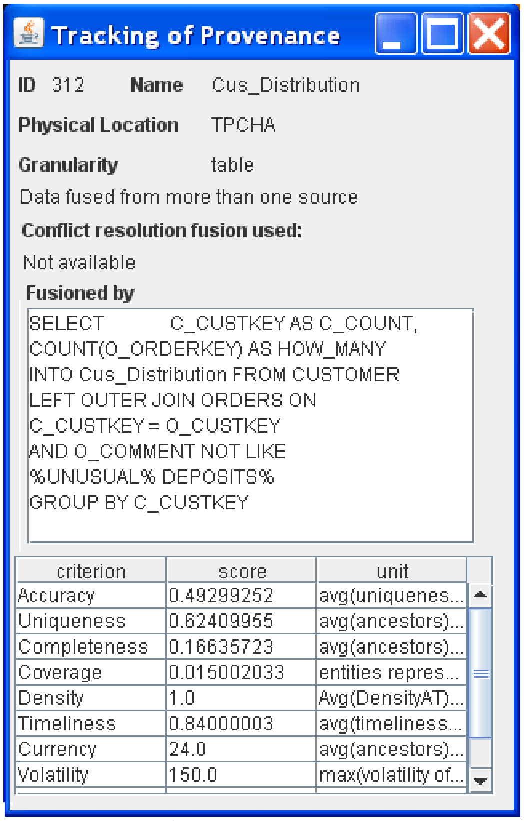


Figure. 5Assessment of Cus_Distribution data quality by the scores aggregation of customer and orders

The following subsection presents some commonly used conflict resolution functions found within the data integration process, and how presenting such information can help users to understand retrieved data.

Enhancing Qualitative Information with the Conflict Resolution Strategies

Previous approaches have developed a number of strategies to resolve conflicts within data fusion [7] [13] [16]. Such information should be taken into account for relying on a data source. Some conflict resolution functions are presented as follows:

Most recent data value: When quality of data is time-related, choosing the most recent value is an option for the solution of conflicting data. When time related data quality dimensions are a priority, then recent value would be preferred.

Most complete data value: Returning the value from the source that contains the fewest NULL values in the attribute in question is recommended if users prefer completeness among other quality properties.

Expert users select data value: The data source has been identified as the best option according with expert users. Therefore, users should take into account that the information retrieved was integrated by a quality dimension called believability, which is particularly relevant in the context of Web.

Selection of the most active data value: In case usability, usefulness, or both are quality properties relevant to the user, this conflict resolution function shall be a good option.

Selection of data value based on the highest quality: The DQM recommends the use of this data value if the quality measure is according to the quality preferences of the data consumer.

Selection of data based on standard aggregation function: The function returns the average, sum, or median value. The DQM recommends this data value as an unbiased and reliable conflict resolution function.

We enhanced the data lineage algorithm we developed in [3] to trace back the conflict resolution functions in order to provide further quality information to users as shown in [1].

During the assessment of data quality by the Data Quality Manager tool, such strategies can be trace back and presented to the user in order to have a better idea what information is being accessed.

The Data Quality Manager prototype provides the physical location, the granularity, the query code or the formula utilized for the data fusion in case of non-atomic data, the provenance tree, and the quality scores of data sources at different levels of granularity.

As we mentioned before, the `pt_profit` query determines how much profit is made on a given line of parts, broken out by supplier nation and year.

The profit is defined as the sum of `[(l_extendedprice*(1-l_discount)) - (ps_supplycost * l_quantity)]` for all line items describing parts in the specified line. Refer to [17] for further detail. Figure 5 presents `pt_profit` as an example of the above mentioned query.

The strategy by which `pt_profit` was selected among other possibilities was because its ancestors where the most active elements within the application of interest. Therefore, the conflict resolution function is presented as “Chosen the most often used data”.

Fig. 6 also presents the scores of the quality properties as a result of assessment by provenance. As we can observe this query is taken information from data sources, which are correct in 82% but not complete (20%), is timely data but very volatile.

The main intension of providing such information is to help users retrieve proper data for operational efficiency and sound decision making.

In the case that a conflict resolution function has been utilized for integrating data, the DQM presents a proper recommendation to users.

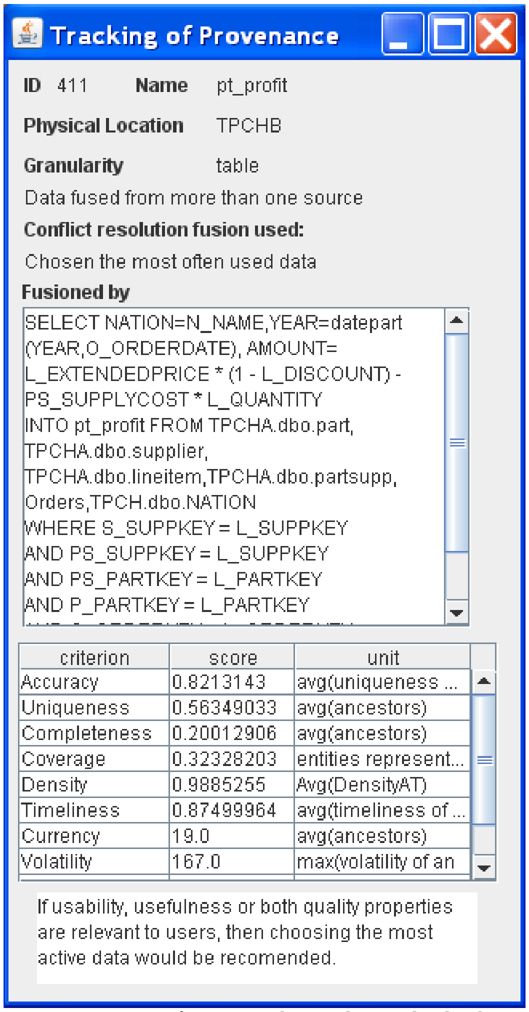


Figure. 6 Assessment of Cus_Distribution data quality by the scores aggregation of customer and orders

Analysis and Ranking of Data Sources

Once the assessment of data has been achieved, the DQM provides the facility to compare data quality of integrated views in order to select the best option. A data quality comparison is presented as follows:

Consider the business question called Market Share, as it was mentioned before, this query determines how the market share of a given nation within a given region has changed over two years for a given part type. There are three possible alternatives to answer the query, called `C_Market_Share`, `D_Market_Share` and `E_Market_Share`. A comparison of such alternatives is possible by the specification of the quality properties of interest. Figure 7 presents accuracy, completeness, and uniqueness as the desired quality properties with their corresponding scores for options C, D, and E.

By default, the DQM is able to apply the proper combination of such methods in order to rank the possible alternatives for the desired global query.

Figure 7 shows assessment and ranking of integrated data, which correspond to the expected outcomes by changing the priority values of chosen quality criteria stated by the user.

We have already explained that the DQM can estimate an overall quality score by providing qualitative information at different levels of granularity, which can vary according to the context specification given by data consumers.

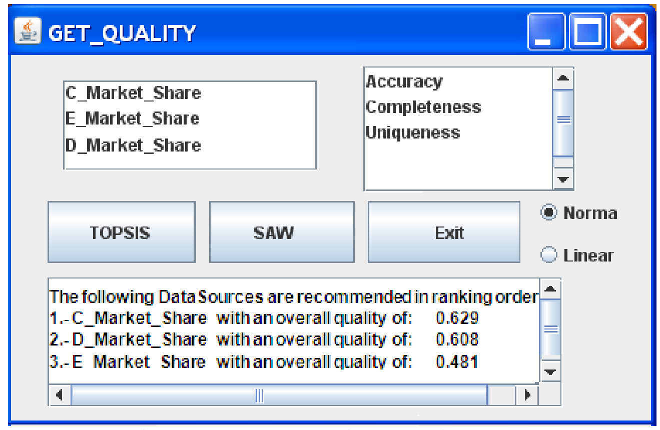


Figure. 7 Specification and execution of ranking of integrated views

As the process of data quality assessment uses a provenance metadata and creates a data quality metadata, in order to analyze data quality changes, the access to these metadata for an ongoing assessment process is required. If data quality assessment is done on a regular basis, users would be able to describe the state of data, to understand problematic data sources, and estimate the cost of data problems to the business.

Assessment of data helps to plan and prioritize data cleansing for improvement, to understand implications of the data quality on newly planned data uses and data driven process before they are put in place [10].

The assessment of data allows the understanding of the current state of data along with the business impact and finding the root causes will lead to a number of activities aimed to prevent data quality problems in addition to correction of current data errors which will be verified by periodic assessments.

E. Business Impact

The enforcement of business rules, the assessment of data quality and the ranking of queries or data sources, let users to identify how root causes affects business.

The data quality scores obtained from the Data Quality Manager inform users which relevant data sources require data cleansing.

The business impact determination varies according to the characteristics of the project, resources, time, and complexity. There are a number of useful techniques such as anecdotes, usage, ranking and prioritization, cost benefit analysis.

In this practical approach we identified ranking of business questions or data sources at different levels of granularity as a very helpful mechanism, for determining the impact of poor data quality on the business.

Ranking of data sources according to its quality allows users to identify which sources of information should be cleaned and which local applications should enhance its business constraints.

F. Data Cleansing

Data Cleansing determines causes of errors and possible treatments. It also creates an audit trail of corrections.

The process of data cleansing requires on the first place identifying the types of errors reducing the data quality then on the second place choosing appropriate methods to automatically detect and remove such anomalies, applying the corresponding methods to the data sources and as a final step examining the results and perform exception handling for the tuples not corrected.

The correct use of metadata has been very useful in order to detect data failing and to establish data profiling and cleansing mechanics. Data consolidation specifications are now built with deep understanding of the actual structure, content, and quality of the data in each source.

Comprehensive data profiling and quality assessment has been a key for success. We started with a comprehensive set of tests, comparing the data between all sources, then we analyzed the discrepancies and look for patterns, if for instance, some time values in two data sources coincide we can trust them and make some corrections on the third one.

Data matching is a very common mechanism to merge and eliminate duplicated rows and keep correct data. At this point we are in enhancing the data matching program. For instance, in the case of text, we have executed a data quality pattern analyzer [35] in the following SQL code:

```
SELECT generate_mask(LINEITEM.ORDERKEY)
AS ORDERKEY_Pattern,
FROM LINEITEM;
```

Table 1 presents the corresponding patterns identified for the O_COMMENT text column.

Table 1 Patterm for O_COMMENT teext column.

O_COMMENT_Pattern
UUUUNNNNNNUUUUUUNN
LLLLNNNNNNWLLLLUNN
UUUUUUNUNUUUNUUNNUU
UUUUNNNNNNUUUUUUUN

After executing the data cleansing processes a certain acceptable level of data quality has been achieved. Therefore, data consumers are able to make effective and informed decisions on the basis of cleansed data at the level of data quality expected. However, as we mentioned before, what is correct today may be completely erroneous tomorrow. In order to maintain the data quality status by preventing new errors from being introduced into the data we require monitoring data integration interfaces and ensuring quality of data conversion and consolidation.

G. Continuous Monitoring and Assessment

After the initial data quality assessment and cleansing, the next step is to ensure that improvements are assigned and implemented. Therefore, we need to plan and implement controls, monitor improvements, and document the results. The successful improvements should be standardized.

Assessing data quality on a regular basis on large volumes of data of a production database is not always viable and technically challenging [10]. The assessment frequency and the level of granularity to assess depend mainly on the objectives stated for the project. A certain level of quality shall be achieved and in the case of that level is inappropriate then assessment and cleansing will be required.

VII. CONCLUSION AND FUTURE WORK

From the existing Data Quality Frameworks, data have always been considered as the product of a primary data source. Therefore, no consideration of derived data has been approached until now. The qualitative information provided to the user contains measures of quality, the original data sources where data come from, and the components of integrated data by considering the process of data integration (i.e. data fusion, data replication, or data transformation) during data quality measurement and assessment. In other words, measuring quality of derived data as part of a Data Quality Framework for multi-database environments has not been addressed before. Very few approaches have considered quality properties at different levels of granularity on databases [12] [14]. Not to mention levels of granularity within derived data.

In the present document, we have shown a practical approach for a proposed Data Quality Framework, where the Data Quality Assessment tool is able to assign quality scores to derived data by considering them as primary data sources, by comparing the available quality scores of its ancestors, or by the aggregation of the quality properties of all its ancestors. Therefore, we presented a new granularity-based assessment classification. Furthermore, qualitative information has been enhanced by including the conflict resolution function and the code or formula utilized for integrating data, depending on the granularity of data along with a brief recommendation to users for trusting data according to the conflict resolution function utilized.

As we mentioned before, data quality degrades during the data integration process [2]. The objective of monitoring these data integration processes is to prevent these errors from getting into the target database. The solution is to design and develop tools between the source and the target data before it is loaded and processed such as the Data Quality Manager for the assessment and ranking of non-atomic data and therefore allow users to be able to make effective decisions by trusting according to the description of qualitative information such as the quality scores, the conflict resolution function, and the quality properties of their ancestors.

The process of determination of cost of data quality by computing the cost to prevent errors, and the cost to correct them is part of our future work.

The process that applies conversion routines to transform data into its preferred and consistent format using both standard and custom business rules stills on development.

We also are planning to extend the presented Data Quality Assessment process to consider semi-structured data.

ACKNOWLEDGMENT

This work was supported by a grant from Dirección General de Asuntos del Personal Académico, UNAM.

REFERENCES

[1] P. Angeles and F. Garcia-Ugalde "Assessing Quality of Derived Non Atomic Data by considering conflict resolution function", First International Conference on Advances in Databases, Knowledge, and Data Applications. 978-0-7695-3550-0/09 © 2009 IEEE DOI 10.1109/DBKDA.2009.10, pp. 81-86, Cancun, Mexico, 2009.

[2] P. Angeles and L. MacKinnon, "Detection and Resolution of Data Inconsistencies, and Data Integration using Data Quality Criteria", Quality in Information and Communications Tech., pp. 87-94, Porto, Portugal, 2004.

[3] P. Angeles and L. MacKinnon, "Tracking Data Provenance with a Shared Metadata", Postgraduate. Research Conference in Electronics, Phot., Comm. and Networks, and Computing Science, pp. 120-121, Lancaster England, 2005.

[4] P. Angeles and L.MacKinnon, "Quality Measurement and Assessment Models Including Data Provenance to Grade Data Sources", Int. Conference on Computer Science and Information Systems", pp. 101-118, Greece, 2005.

[5] P. Angeles, "Management of Data Quality when Integrating Data with Known Provenance", PhD Thesis, Heriot-Watt University, Edinburgh, UK, April 2007.

[6] D. Ballou, G. Tayi, "Examining Data Quality", Communications of the ACM, vol. 41,no.2, pp.54-57, 1998.

[7] J. Bleiholder. Declarative Data Fusion, Syntax, Semantics, and Implementation. Advances in DB and I S, Estonia, 2005, pp. 58-73, 2005

[8] M. Burgess, W. Gray, and N. Fiddian, "A Flexible Quality Framework For Use Within Information Retrieval", Int. Conference on IQ,Cambridge, MA, USA, 2003.

[9] M. Gertz and I. Schmitt, "Data Integration Techniques Based on Data Quality Aspects", 3rd National Workshop on Federal Databases, 1998.

[10] A. Maydanchik, Data Quality Asseessment, Data Quality for Practitioners Series, Technics Publications New Jersey ISBN 978-0-9771400-2-2, 2007.

[11] A. Motro and I. Rakov I, "Estimating the Quality of DB", Int. Conference on Flexible Query Answering Systems, pp. 298-307,Springer-Verlag, Germany, 1998.

[12] F. Naumann, "Quality-Driven Query Answering for Integrated IS", Lecture Notes in Computer Sciences LNCS 2261, Springer Verlag, 2002.

[13] F. Naumann, A. Bilke, J. Bleiholder, M. Weis "Data Fusion in Three Steps: Resolving Inconsistencies at Schema, Tuple and Value-level, IEEE Data Engineering Bulletin 29(2):21-31, 2006.

[14] L. Pipino, W.L. Yang and R. Wang, "Data Quality Assessment", Communications of the ACM, Vol. 44 no. 4e, pp.211-218, 2002.

[15] M. Scannapieco, A. Virgillito, et.al. "The DaQuinCIS Architecture: a Platform for Exchanging and Improving DQ in Cooperative IS", Information Systems, Elsevier, pp. 551-582, 2004.

[16] Schallehn E., Sattler Kai-Uwe, Saake G., Efficient similarity-based operations for data integration Data & Knowledge Engineering, Vol. 48, 3, 2004, Pages 361-387

[17] TPCH, TPC Benchmark ™ H, Standard Specification Revision 2.3.0 Transaction Processing Performance Council, http://www.tpc.org , 2006, (date information as accessed by the author citing the references, e.g. 23 Sept. 2009.)

[18] R. Wang, "A Product Perspective on Total Data Quality Management", Communications of the ACM, vol. 41, no. 2, pp.58-65, 1998.

[19] D. McGilvray Executing Data Quality Projects Ten Steps to Quality Data and Trusted Information, ISBN 978-0-12-374369-5, Morgan Kaufman, Publishers, 2008.

[20] M. Scannapieco, C. Batini, "Completeness in the Relational Model: A Comprehensive Framework", Research Paper, in Proceedings of the 9h International Conference on Information Quality (ICIQ-04, Cambridge, MA, USA, November 2004.

[21] L. Young and D. Strong "Knowing-Why about Data Processes and Data Quality", Journal of Management Information Systems, Vol. 20, No. 3, pp. 13 – 39. 2004.

[22] T. Anderson , The Penalties of Poor Data, Immedia smart targeted solutions., http://www.goimmedia.com/ArticlesWhitepapers/ThePenaltiesofPoorData.aspx , (date information as accessed by the author citing the references, e.g. 23 Sept. 2009.)

[23] D. Loshin, Enterprise Knowledge Management, The Data Quality Approach, 2007.

[24] New York Times, December 1, 2002, by Jennifer Bayot.

[25] New York Times, January 19, 2003, by The Associated Press.

[26] New York Times, February 28, 2003, by Karen W. Arenson.

[27] Y.Lee, D. Strong, "Knowing-Why about Data Processes and Data Quality", Journal of Management Information Systems, Vol. 20, No. 3, pp. 13 – 39. 2004.

[28] The MIT Total Data Quality Management web site, http://web.mit.edu/tdqm/, (date information as accessed by the author citing the references, e.g. 23 Sept. 2009.)

[29] P. Anokhin, A. Motro, "Fusionplex: Resolution of Data Inconsistencies in the Integration of Heterogeneous Information Sources", Technical Report ISE-TR-03-06, Information and Software Engineering Dept., George Mason Univ., Fairfax, Virginia, 2003.

[30] J. Cavano, "A Framewok for the Measurement of Sotware Quality",Rome Air Development Center, James A. McCall, General Electric Company (1978),pp.133-139.

[31] K.T. Huang, Y.W. Lee, R.Y. Wang, Quality Information and Knowledge Management,Prentice Hall PTR Upper Saddle River, NJ, USA, ISBN:0-13-010141-9.

[32] A.F. Karr, A.OP. Sanil, D.L.Banks ," Data Quality: A Statistical Perspective", Technical Report 151, March 2005, National Institute of Statistical Sciences.

[33] T. C. Redman, "Data Quality for the Information Age", Boston, MA., London : Artech House, 1996, ISBN:0890068836.

[34] Wang R. Y., Strong D.M. "Beyond accuracy: What data quality means to Data Consumers", Journal of Management of Information Systems, vol. 12, no 4 1996, pp. 5 -33.

[35] Data Quality Pro Forum, http://www.dataqualitypro.com/(date information as accessed by the author citing the references, e.g. 23 Sept. 2009.)

Composition of context aware mobile services using a semantic context model

João Paulo Sousa
Departamento de Informática e Comunicações
Instituto Politécnico de Bragança
Bragança, Portugal
jpaulo@ipb.pt

Eurico Carrapatoso
Faculdade de Engenharias/INESC Porto
Universidade do Porto
Porto, Portugal
emc@fe.up.pt

Benjamin Fonseca
CITAB/Universidade de Trás-os-Montes e Alto Douro
Vila Real, Portugal
benjaf@utad.pt

Maria da Graça Campos Pimentel
Departamento de Ciências de Computação
Universidade de São Paulo
São Carlos-SP, Brazil
mgp@icmc.usp.br

Renato de Freitas Bulcão-Neto
Innolution Sistemas de Informática
Ribeirão Preto-SP, Brazil
rbulcao@innolution.com.br

***Abstract*— Context-awareness has been regarded as an important feature for mobile services. However, only a few services are sensible to context and the features that are context-aware are still limited. Composition of Web services has received much interest in business-to-business or enterprise application, but not so much interest in business-to-consumer applications. This paper presents iCas, a novel architecture that enables the creation of context-aware services on the fly, and discusses its main components. We compare our approach with similar systems and point out the main differences and advantages. To explore context-awareness to support service composition, iCas uses SeCoM, a semantic model to represent context. The main parts of this model are explained as well the advantages of using a semantic model to represent context. We also describe the use of our approach in an university campus to provide pedagogical features and assist the socio-pedagogical interaction of various types of users.**

Keywords: Context-aware, Services composition, Semantic Web, Web Services

I. INTRODUCTION

It is predictable that in the near future the network mobile environment will be characterized by interaction between services and that those services will be provided to users dynamically and transparently. In this scenario, the use of captured contextual information related to issues such as location, current activities, objects in the neighbourhood and device features plays a crucial role in the simplification of the interaction between humans and the digital world.

Often users only assume the role of consumers of services provided by third parties. For those users a set of useful services and information is provided, but they are aimed at a general market, leaving aside users that would like to take advantage of more personalized services. This paper proposes and describes a service oriented open infrastructure for a mobile network environment. We call this architecture iCas and it allows a user to receive in his mobile device (e.g. PDA, netbook, notebook) context-aware information (e.g. location, time, neighborhood, user profile) and have a set of useful services that are sensitive to his current context. The user can also compose services dynamically in real time to create a new highly personalized enviremententwith more features and use or share it as many times as he wants [1].

The remainder of this paper is structured as follows: section 2 discusses related work, section 3 presents some definitions of context, and section 4 introduces the SeCoM semantic model to describe and to provide reasoning about context. Section 5 discusses the several approaches to composing Web Services and the main innovations of our proposal, followed by the description of the OWL-S ontology to support semantic Web Services. Section 6 presents the iCas, a Service Oriented Architecture (SOA) and describes the details of each of itscomponent. Section 7 presentsa scenario for using iCas, a university campus, where iCas will be used to allow users to compose in a had-hoc way new services for enhancing everyday campus life. Section 8 describes the first performance evaluation. Finally, we provide some conclusions and suggestions future work, in section 9.

II. RELATED WORK

A number of context-aware systems has been developed to demonstrate the usefulness of context-aware technology, such as ParcTab [2], which was one of the first systems to offer a general context-aware framework and ContextToolkit [3], which presents a modular context-aware framework with reusable components. Which allows programmers to build more easily interactive context-aware systems based on sensors. These systems donot have an open context model because often the context is described in an object-oriented basis and so the information is strongly coupled tothe programming model.

More recently several studies appeared to support context-aware composition of services, one more generic and others dedicated to mobile environments [4][5][6][7][8].

In [4] the authors present a distributed architecture and associated protocols for service composition in mobile environments. This study emphasizes some factors that allow the composition of services in ad-hoc networks such as mobility, dynamic changing service topology, device heterogeneity, fault tolerance and reliability.

In [5] the authors propose a framework for dynamic composition of context-aware mobile services. The main features are service adaptation to devices and networks, and service adaptation to the user preferences and user location. However the study does not specify which approach is used to compose new services.

SOCAM [6] presents a middleware architecture for rapidly building context-aware services. It provides support for discovering, acquiring, interpreting and accessing context information. It also presents one of the first ontologies that define the main classes of context: person, location, activity and computer entity. Nevertheless, this architecture does not allow the composition of services. MyCampus [7] is a semantic web environment that uses agents that are able to find context information to improve users' campus life. The MyCampus architecture is composed by eWallets (static knowledge containers), which support automated discovery and access to the context. The users can subscribe task-specific agents to assist them in different context tasks using the semantic information in eWallets. These agents are able to discover, execute and compose automatic semantic Web services using the Semantic Markup for Web Services (OWL-S) [9].

In [8] the authors present CACS a framework that enables context-aware composition of Web Services. This framework supports capability matches and goal-driven composition services flow. The CACS architecture uses software agents to discover, compose, select, and automatically execute Web Services using OWL-S.

In [4][5][7][8] we saw that these systems do nothave an open model to describe context, which causes some limitations on sharing context knowledge and context reasoning with external systems. The studies[4][5][8] present architectures that support the automatic composition of services. The user makes a request to the architecture, most of the times to a software agent, whichcollects context information and tries to find the most suitable service, which agrees with the request description. If the agent doesnot find the service or it doesnot exist, then the software agent decomposes the request into multiple sub-goals in order to find the matching services.

In all the cases that use automatic composition, it is a hard task to maintain the details about the rules of services' invocation. These approaches also do not have an open model to describe context, which causes some limitations regarding the sharing of context knowledge and context reasoning with external systems.

III. CONTEXTUAL INFORMATION

The development of an architecture that uses context information requires the perception of the meaning of context and how it can be used. A phenomenon that is observed when someone is asked about what context is that most of the people understand what it is, but they feel that it is hard to explain. For this reason many timescontext definitions are done by enumeration of examples or by choosing synonyms for the context.

The term context was introduced for the first time in [10], referring it location, people, hosts and accessible devices nearby, as well as changes to such things. On [11], the authors define context as location, people in the neighborhood of the user, time and temperature, among others. In [12]context is defined as being the user location,

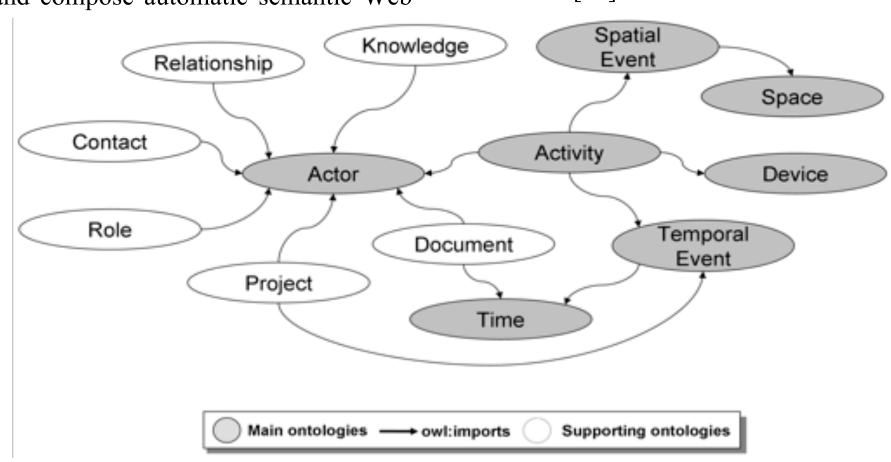


Figure. 1 An overview of the SeCoM model [16].

environment, identity and time information. In [13] the authors have the following interpretation of context: “Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the applications themselves”. The authors in [14] present another understanding of context. They define it as everything that affects the computation except the explicit input and output data.

There are more context definitions, some described by examples, others described generically and some other in a more explicit way. After we made the review about the meaning of context, we understood context as all the information captured in a non-explicit way and used to create dynamic rules that change the way that services and information are provided to an actor. An actor can be a human or a software agent.

IV. The use of a Semantic Model

Contextual information models based on ontologies have been explored in several architectures that support context-aware services (e.g. [6][15][16]). These models allow the cooperation among objects and the discovering, acquisition, inference, and distribution of contextual information. An ontology is defined by R. Gruber as an explicit and formal specification of a conceptualisation of a domain of interest [17]. Ontologies consist of concepts (known as classes), relations (properties), instances and axioms, and on the computing context.Ontologies provide a shared understanding between applications of a domain, typically the common sense about that domain.

To describe the context, we decided to use the semantic model SeCoM (Semantic Context Model), presented in [16].

The use of a semantic model brings about several advantages:

- the possibility of having a high degree of expressiveness and formalism to represent concepts and relations in a context-aware scenario; it allows reasoning about context;
- the use of a semantic information context model, based on Semantic Web standards, makes the exchange, reuse and sharing of context information between context-aware applications easier;
- it decouples the information context model from the programming model, unlike some architectures presented in the section II.

SeCoM is composed of six main ontologies: Actor, Activity, Space, Spatial Event, Temporal Event, Device, Time, and six support ontologies, Contact, Relationship, Role, Project, Document, Knowledge. Fig. 1 shows the SeCoM ontologies and their relationships.

A. *The SeCoM Model: An Overview*

Considering context modelling, we have developed the Semantic Context Model (SeCoM) [16, 18], which represents the semantics of context information through a set of semantic web ontologies. From the perspective of a context information model, the following is the list of SeCoM's main characteristics:

- it is an effort towards a domain-independent model for context-aware computing ;
- it models classical types of context information such as who (identity), where (location), when (time), what (event and activity) and how (device) [19];
- it is semantic-oriented with high level of expressiveness and formality borrowed from the Description Logics (DL) [20], which is a mature knowledge representation technique representing a subset of first-order logic;
- it is based on ontologies as formalism of context information representation, which is, in turn, based on DL expressiveness and decidability;
- it is a modular model, where each type of context information is represented in a particular ontology to facilitate both its reuse and extension;
- it reuses concepts from general consensusandstandardized Semantic Web ontologies;
- it allows inference of new facts from previous context information due to its ontological semantics;
- it uses Semantic Web standards for representing the structural, semantic and logic views of context information such as Resource Description Framework (RDF) [21] and Web Ontology Language(OWL) [22];
- it is a two-layered context information model, whichfacilitates the task of an application developer to reuse and/or extend the most general concepts of SeCoM.

B. *The SeCoM model: A Detailed View*

The main ontologies composing the SeCoM context information model are briefly presented next. Further information on the SeCoM model found elsewhere [16, 18, 23].

1) ACTOR ontology: it models the profile of entities performing actions in an ubiquitous computing environment such as people, groups and organizations.

2) TIME ontology: it models temporal information in terms of time instants and time intervals (two or more not null time instants), relations between time instants and intervals (temporal mereology), relations between time intervals (mainly based on Allen's Temporal Algebra [24]), and calendar and clock information (time duration, day of week, month of year, etc.).

3) TEMPORAL EVENT ontology: it models events with temporal extensions such as instant or interval events. It is an extension of the Time ontology because temporal events are defined assubclasses of the class *time:TemporalThing*. In other words, it is able to represent temporal methology between instant and interval events, and temporal relations between interval events. In addition, this ontology also represents information about periodic temporal events such as the frequency of an event, or even the time duration between occurrences of an event.

4) SPACE ontology: it describes the whereabouts of actors. It models virtual and real-world indoor (e.g. *Room*) and outdoor (e.g. *Street*) places, mereological (e.g. *spatiallyContains*) and spatial (e.g. *isSpatiallyConnectedTo*) relations between places, geographic coordinates (e.g. *latitude*) and *directions* (e.g. *north*) and administrative regions (e.g. *City*).

5) SPATIAL EVENT ontology: it models events with spatial extensions called spatial events, which are subclasses of *spl:SpatialThing* defined in the Space ontology. Spatial events can be represented by two main disjoint subclasses: physical events, which are those occuring in a physical location (e.g. entrance in a meeting room), and virtual events, which include those occuring in a virtual location (e.g. entrance in a chat room). In general, both physical and virtual spatial events inherit all properties, relations and axioms from the classes *spc:PhysicalLocation* and *spc:VirtualLocation*, respectively.

6) DEVICE ontology: it describes computational devices that can be used in ubiquitous computing interactions. The main concern is to represent those devices in terms of their hardware and software platforms, mereological relations between their components, and mobile computing aspects needed for context-aware computing. In general, it models information about storage and battery capacity, multimedia support, wireless and wired network connectivity, operating systems and browsers supported, virtual machines installed, among others.

7) ACTIVITY ontology: it describes activities as sets of spatiotemporal events including the corresponding actors and devices involved in. Thus, this ontology directly imports the Actor, Spatial Event, Temporal Event and Device ontologies, as depicted in Fig. 2. Being modeled as spatiotemporal events, activities reuse the same attributes and relations of both spatial and temporal events. In other words, it is possible to interrelate activities in terms of mereological and spatial relations between their physical/virtual locations, or even in terms of temporal relations between their corresponding time instants and intervals. Besides, it also models activities as of two disjoint types: impromptu and scheduled. The former represents activities occuring in an informal manner (e.g. cocktail meetings), whereas the latter represents activities planned in terms of time and space (e.g. lectures at a conference room). The following is an RDF excerpt of a Computer Science Conference activity represented as a scheduled activity occuring at the "DVR-001" Da Vinci room, which is located on the Conference floor at a university. CS conference started at 10 am on March 7, 2009, and it took two hours long. Activities' participants are described by means of the property actvy:hasParticipant. The actvy: prefix is used to represent the XML namespace for the Activity ontology. In terms of temporal and spatial reasoning, a reasoner could infer that this computing conference still took place at 11 am on the Conference floor.

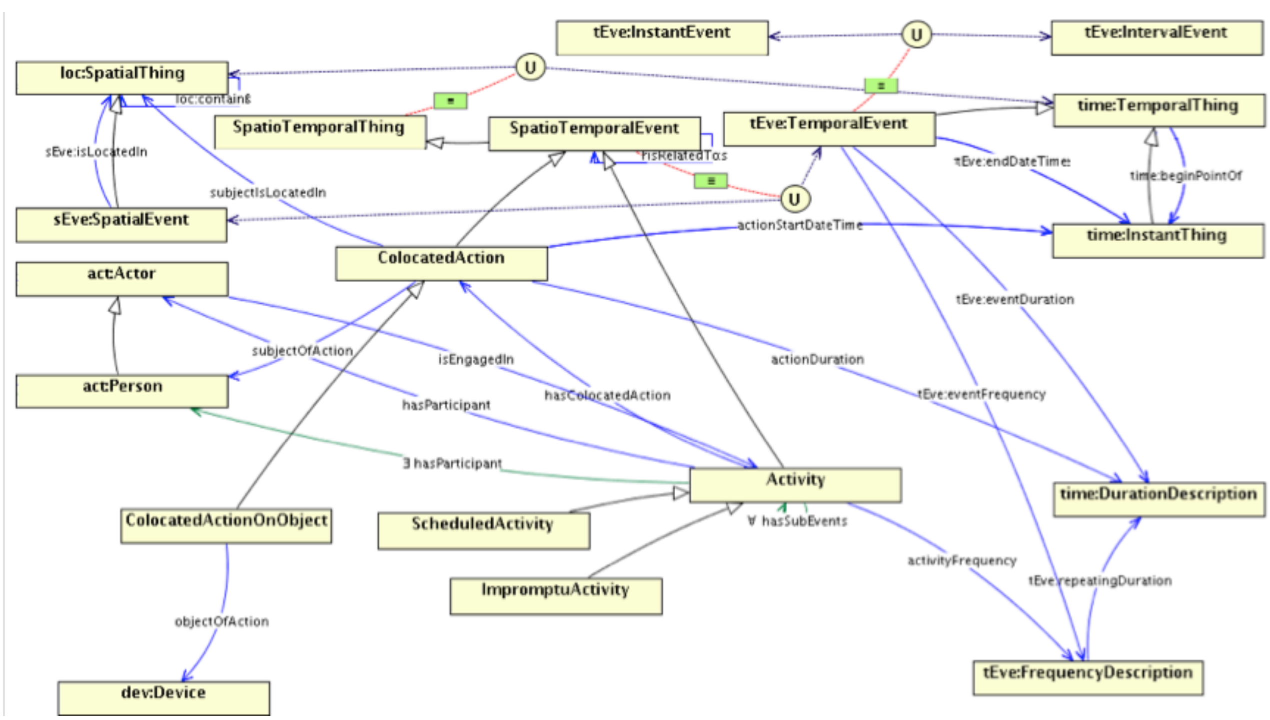


Figure. 2 The Activity ontology.

```
<actvy:CSConference rdf:ID="cmeeting19">
<rdf:type rdf:resource=
"&actvy;#ScheduledActivity"/>
<actvy:hasParticipant rdf:resource="#person19"/>
<sEve:isLocatedIn rdf:resource="#room82"/>
<time:beginPointOf rdf:resource="#bpo67"/>
<time:intervalDurationDescriptionDataType
       rdf:datatype="&xsd;#duration">PT2H
</time:intervalDurationDescriptionDataType>
</actvy:CSConference>
<act:Person rdf:ID="person19">
<act:hasName>Claus Ana</act:hasName>
</act:Person>
<spc:DaVinciRoom rdf:ID="room82">
<rdf:type rdf:resource="&spc;#Room"/>
<spc:placeName>DVR-001</spc:placeName>
<spc:isSpatiallyPartOf rdf:resource="#floor4"/>
</spc:DaVinciRoom>
<spc:ConferenceFloor rdf:ID="floor4">
<rdf:type rdf:resource="&spc;#Floor"/>
<spc:placeName>Conference floor
</spc:placeName>
<spc:isSpatiallyPartOf df:resource="#ipb"/>
</spc:ConferenceFloor>
<time:InstantThing rdf:ID="bpo67">
<time:instantCalendarClockDataType
        rdf:datatype="&xsd;#dateTime">
        2007-03-07T10:00
</time:instantCalendarClockDataType>
</time:InstantThing>
```

V. WEB SERVICES COMPOSITION

The composition of services allows developers and users to create new services or applications, based on a Service Oriented Architecture (SOA) that supports description, discovery and communication. One of the most used SOA technologies is Web Services, due to the advantages already known to the scientific community [25][26][27].

Web Services have often been used for the composition of services. Nowadays there are six approaches to the Web Services composition [28]: WSBPEL [29], Semantic Markup for Web Services (OWL-S) [30], Web Components [31], Algebraic Process Composition [32], Petri Nets [33] and Model Checking and Finite-States Machines [34]. The previous approaches intended to solve the problems found in services composition such as syntax and semantic verification, resource reservation, QoS or deadlocks. In [28] and [35] the authors compare several solutions, based on characteristics such as automatic composition, composition verification, scalability, goal satisfaction, connectivity and non-functional properties.

When the purpose is to implement the composition of mobile services, we have to consider some concerns such as the complexity of the services to be built. For this purpose, wemust find a compromise between simplicity in service creation and flexibility:a more flexible service requires more complex rules and probably specific technical knowledge. In this case the simplicity offered to end users is lost.

To achieve this goal, we chose to compose services in an interactive way: the user gradually generates the composition with ad-hoc forward or backward selection of services. Using this approach for composing Web services requires that they understand their features and how they interact together. The Web Services Definition Language (WSDL) [36] specifies a standard way to describe the interfaces of a Web Service at the syntactic level. However, WSDL does not support the semantic description of services. OWL-S has appeared to fulfill this limitation and uses the OWL language to describe Web Services. OWL-S provides Web services with a set of markup language constructs for describing the

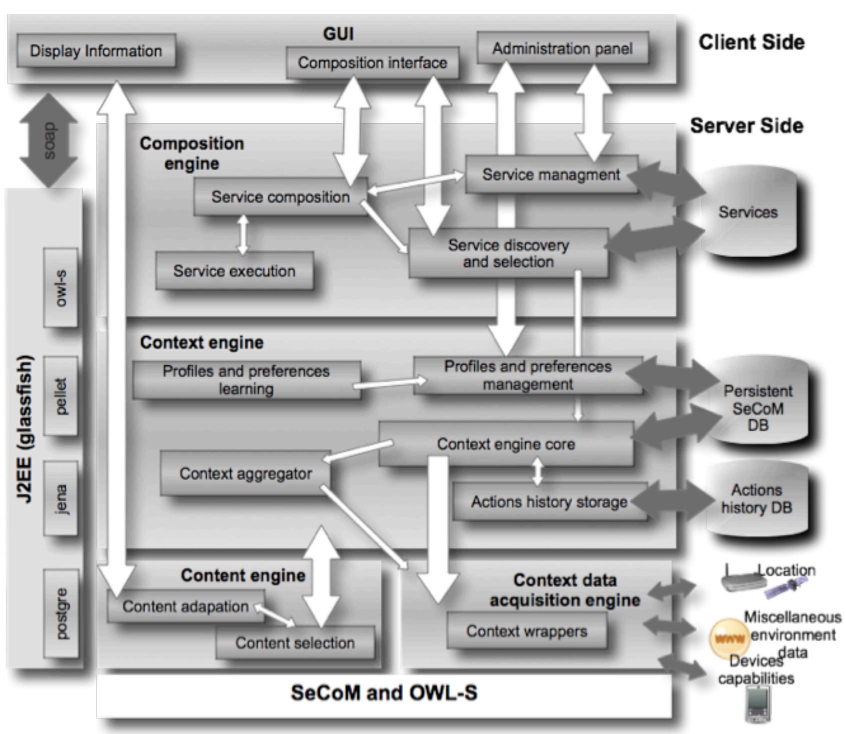


Figure. 3 Overview of iCas architecture.

properties and capabilities in anunambiguous interpretable form to the software agents. OWL-S is a framework that enables automatic discovery and matchmaking tasks, and composition and execution of Web Services.

OWL-S consists of the following classes: ServiceProfile - specifies how the services are announced to the world; ServiceModel - specifies how to interact with the service; ServiceGrounding - specifies the details of how an agent can access the service.

VI. PROPOSED ARCHITECTURE AND IMPLEMENTATION

To support the composition of context-aware services on the fly and provide context-aware information to the users, we propose a Service Oriented Architecture (SOA) based on ontologies. We divide the architecture into four essential engines to explore the potential of context, showed in Fig. 3.

When a user choosesthe service composition IDE, the service discovery component gets the preferences, parameters and interests. With this information and the OWL-S services descriptions, the service discovery and selection selects the services from the service repository to perform a context-based selection, and then delivers it as a list to the IDE.

When a user starts a composition, maybe he knows clearly which tasks he wants to achieve with the composition or perhaps he starts to compose, choosing compatible services that can suggest the creation of a new service. In either situation the service composition is an ongoing process, where the user can add or remove services interactively.

Each time a service is selected to be part of the composition, the service discovery and selection module searches for services (Fig. 4) using data collected from the context engine core and returns further possibilities based on the current context and user policies. The search and selection is only possible due to the OWL-S service description, which allows creating relationships with other ontologies that can describe details about a service type and its features.

The search is performed using the description of the ServiceProfile class, which contains what the services can do, and specifies the input/output types, preconditions and effects. The first selection of services is carriedusing the ServiceProfile hierarchies, which choose the services from a particular category. Then a matching is performed, selecting the services whose input is syntactically compatible with the output of the current service.

Finally a scoring is carried out using the weights of the evaluation parameters defined in the ServiceProfile and a particular evaluation policy, which depends on the service category.

The ongoing user composition is supported by the service composition function, which generates a workflow of services calls. Fig 5 shows an overview of the interactions between the components from the several engines and the GUI , when a composition is accomplished.

By the time that a user finishes the composition, the entity service composition has created a composite service that contains a workflow. This workflow is a composite service that has the three key descriptions of an OWL-S service: service profile, grounding and model, as mentioned in the end of section V. This newly composed service can be saved, executed or used in another service composition task. To store the service, the service composer uses the service management component, and to execute the service it calls the service execution component.

The service management component deals with the services stored in the services container, providing operations such as adding, removing and sharing services using the policies properties. The service container only stores the OWL-S description of the service (service profile,

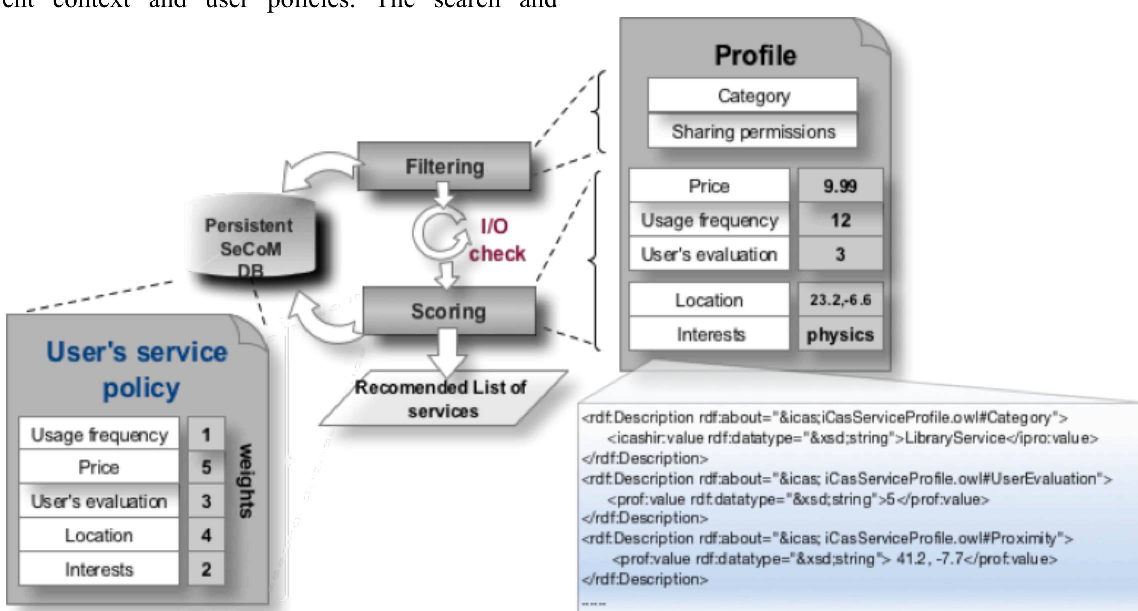


Figure. 4 Service selection mechanism.

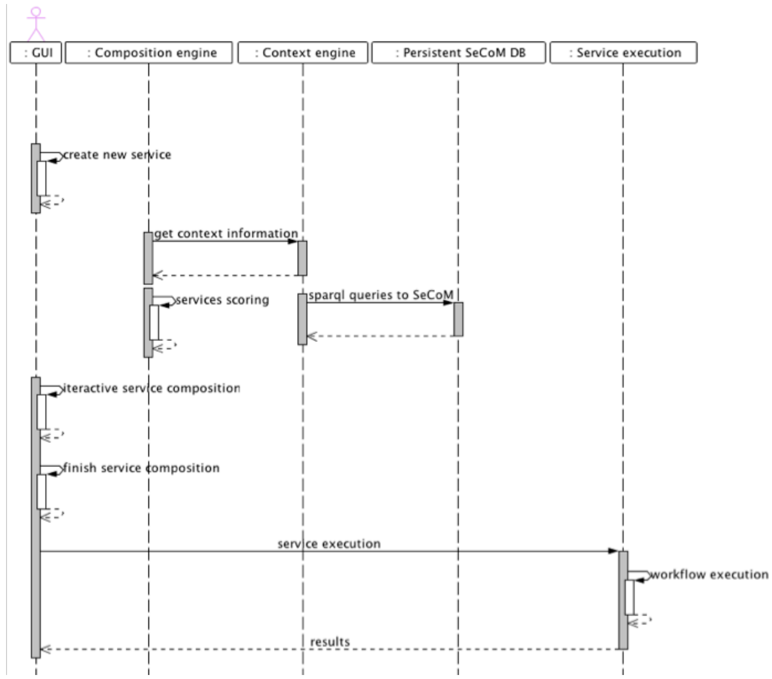


Figure. 5 Composition sequence diagram.

model and grounding). The service functionality is still provided by a third party (e.g. e-learning platform Web service).

The service execution module, using the OWL-S API, provides an execution engine to invoke atomic processes described by WSDL or Universal Plug and Play (UPnP) [37] groundings, and composite processes that use control constructs sequences, unordered, and split. All execution processes that depend on conditional statements, such as if-then-else and repeat-until, are not supported by the API. When the service execution promotes a composition, it follows a workflow to call each individual service and exchange data between them, according to the flow defined by the user.

The context engine is responsible for managing all related context data and for reasoning about context. All context information is stored in a permanent OWL ontology storage system. The context engine core uses the Jena API to store the RDF models of SeCoM using a Postgre DB. This engine is also responsible for extracting knowledge from the SeCoM ontology, using RDF Query Language and Protocol (SPARQL) [38] queries and for making inferences to derive additional statements that are not described explicitly in the SeCoM model. The following code is a SPARQL query to the persistent ontologies, to get all the events related with the Computing subject and their location.

```
PREFIX rdf: <http://w3.org/1999/02/rdf-syntax-ns#>
PREFIX acti:<http://icas.ipb.pt/activity.owl#>
PREFIX spac:<http://icas.ipb.pt/spatial.owl#>
SELECT DISTINCT ?event ?subjectIsLocatedIn ?hasName
 WHERE {
  ?subjectIsLocatedIn spac:hasName ?hasName
  ?hasColocatAction acti:subIsLocatIn ?subIsLocatIn
  ?event acti:hasColocatAction ?hasColocatAction
  ?event a acti:ScheduledActivity
  ?event acti:hasSummary ?hasSummary
  ?event acti:validationStatus true
  FILTER regex(?hasSummary, "Computing")
 }
```

Using OWL's capabilities also enables to make inferences using the Pellet reasoner, (e.g. "if a user is located in the library, he is in university campus", or if a user has interests in "ontologies", and because ontologies has a transitive properties with "semantic web" and this one also related with "context-awareness", hence the user is also interested in "context-awareness").

The context aggregators keep in memory (non-persistent) highly changing dynamic data that is captured from various sources related to an entity (e.g. user, object). For each entity an instance is created that relates that entity with data from the sources (e.g. user's location and data sensor). This component moves the computational charge caused by the frequent data updates into the persistent ontology.

The profiles and preferences management component is responsible for managing the explicit user profile and interests information. Using the administration panel this component allows the user or administrator to manage explicit context such as insert, update and remove profile parameters and user preferences.

The actions history storage component captures each action performed by the context engine core and stores it in the actions history DB. The main actions are search, insert, update and remove, and they are stored in the following format: Action + target Triplet (e.g. update: Bob isMemberOf the Sciences Students Group).

The profile and preferences learning component can change preferences and profile data using historic information of user actions (e.g. if a student queries many times a particular book in the library services, the theme category of that book is added to the hasInterestesIn property of the knowledge ontology). The profile and preferences

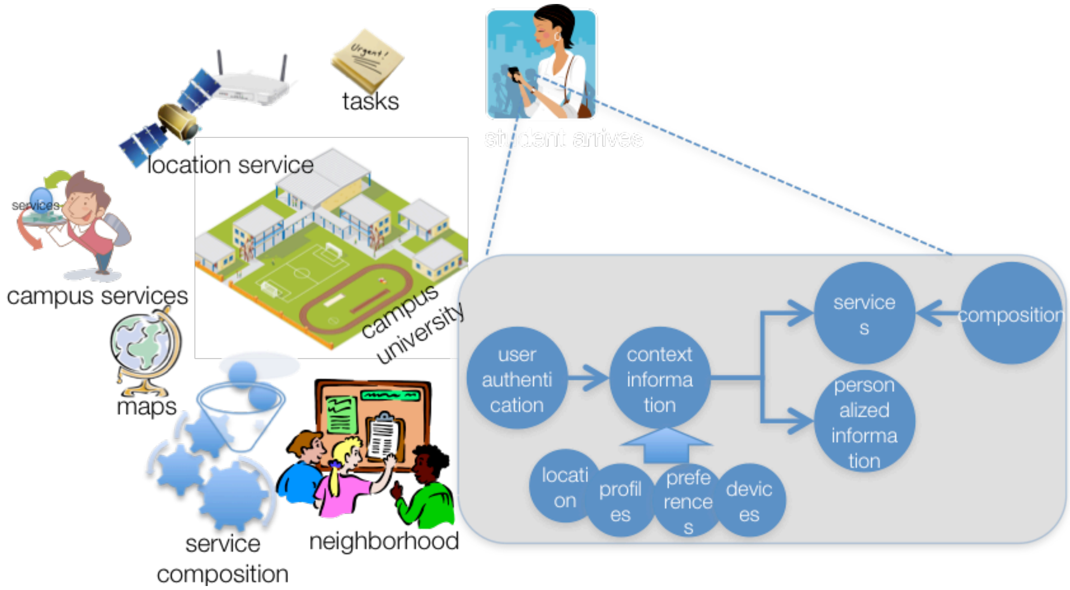


Figure. 6 iCas usage scenario in a university campus.

learning is an independent component. It searches for particular actions stored in the actions history DB, and counts the number of times that an action appears and, accordingly, changes specific parameters defined to be learned. Although this is not an optimal approach, a good solution can only be achieved with a large-scale utilization of iCas architecture and the collecting of user feedback. In the future this mechanism may also evolve to an AI algorithm, searching for patterns in the database.

The context data acquisition engine collects data from several sources, such as location devices, sensors and external services, and prepares the data to be used by the content engine and context engine (e.g. convert units values from a data sensor, or transform the coordinates user's location to a referential location (room 2.1)).

The content engine is composed by two components: the content selection is a timer function that periodically selects the user interests information from the context engine and delivers it to the content adaptation module for transformation. To be able to consult information in arbitrary devices, the information content must be provided in a device-independent way. iCas provides the context information as RSS feeds that are adapted by the content adaptation component. To do that this component adapts the information to the user's device features, using XHTML Modularization [39].

The iCas system is implemented integrally in Java (JDK 1.6.0). The iCas middleware architecture is composed of:

- Composition engine and context information system: Glassfish v2, JAX-WS 2.1, JAXB 2.1, Jena 2.5.4 and OWL-S 1.1.
- Context, profiles and preferences management DB: Postgre 8.2.8.
- Actions history storage management DB: Postgre 8.2.8.
- Ontologies models: SeCoM and OWL-S.

All four engines are implemented in the Glassfish v2 application server, which provides the functions to the GUI client through HTTP, as Web Services. This configuration was chosen to support the ad-hoc composition of services in mobile devices, bringing the reasoner's computational requirements to the server side.

VII. EXAMPLE OF APPLICATION

We have chosen a university campus as a scenario for using iCas (Fig. 6). This architecture aims the support students and teachers in their campus life, helping them to keep updated and improve their social and pedagogical interaction.

When a student arrives at the campus and connects his mobile device to the wireless network he will have to authenticate. This authentication is used to identify the user in a WiFi campus system and in the iCas architecture.

The campus university already has a location system based on the wireless network, which is used to locate the users inside the campus. Besides the service location, the campus also has other services that can provide useful information integrated to the iCas system. Some of the most important services are: an e-learning platform that provides news about lessons, classes contents and others pedagogical information; library services and administrative services.

To implement a scenario we developed an iCas Client application. Fig. 7, shows the iCas client adapted from the Web Service Composer application [40], under the terms of the GNU Lesser General Public License. The main features of iCas consist of providing context-aware information and the dynamic composition of services. For this purpose the user's GUI client has four panels: informative, services composition, maps and administration.

In the information panel the user can access campus information based on his context (e.g., activities, events,

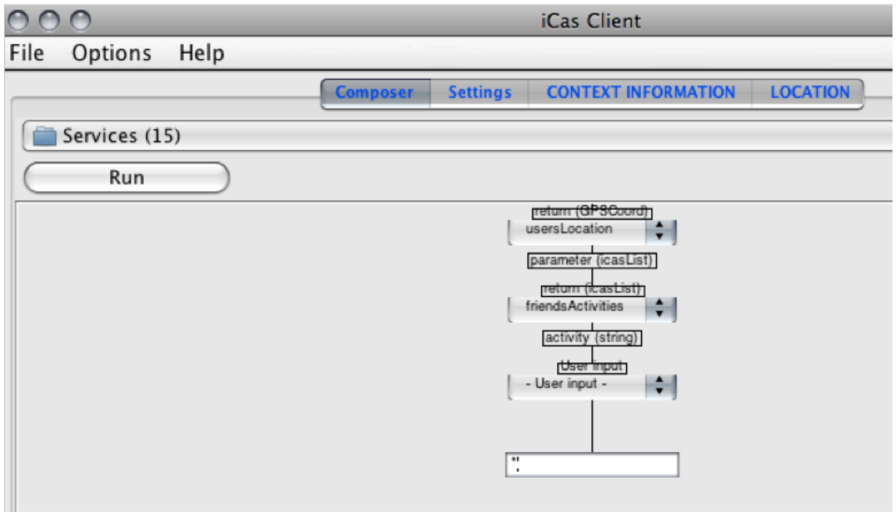


Figure. 7 iCas client prototype – composition panel

news). To compose services in an ad-hoc away the user can use the services composition panel. If the user uses any service that has location output format, information will appear on the maps panel. Any task related with administration, such as changing user profile data and other explicit information, has to be done in the administration panel.

A typical example of this usage scenario is the Friends' Awareness Location Service, in which the user combines the following set of services to get information about the activities of friends that are located in the campus: User's Activities – information gathered by analyzing the user's profile and Users Location Service – provides users locations based on the information gathered on the campus location system aforementioned.

Fig. 8 shows the previous composition built in the composer panel, with Friends' Activities service and Users Location service selected.

When a user starts to compose a new service he selects the composition panel and a list of the available services is presented to him, sorted by the service selection mechanism shown in Fig 5, and described in section VI. During the search for available services he sees two services that might fit his needs: the Friends' Activities service and Users Location service. So, he starts to compose the services and chooses first the Friends' Activities service but when he tries to select the next service to join, he realizes that the User's Location doesnot appear in the list of available services. This happens because itoccurred an incompatible matching betweenthe output of the User's Location (GPSCoord) and the input of Friend's Activities (Activity). So he starts again, selecting first the User's Location service. Next he can find the Friend's Activities service in the list of available service.

The service is available to join for composition, because

Figure. 8 iCas client prototype – maps panel

Table 1 Times to insert context usingthe Context Engine Core.

Number of entries	Add People (ms)	Add Places (ms)	Add Devices (ms)	Add Schedule Activities (ms)	Sum (ms)	Add All (ms)	Upate User Location (ms)
500	2688	3451	2562	4804	13506	19905	9560
1000	5251	6960	5668	10694	28573	34897	35036
2000	9822	14014	11755	20237	55827	66283	41036
5000	24555	34071	26363	51972	136961	154992	97773
10000	52232	71400	57125	110720	291477	300574	130370

now the output and input parameters are syntactically compatible. Next the user writes a wildcard in the input box to know all the activities of his friends. The output (users activities and location) of the service composition is presented in Fig. 8. In the end the user also can save the new composed service to use next time or share it with other users.

VIII. PERFORMANCE EVALUATION

The implementation presented in the section VI is ongoing work. To get the first performance evaluation, we tested some components that we consider critical to evaluate the performance of iCas architecture.

A. *Testing Scenario*

As seen in the previous section a limited client prototype was implemented which despite being tested by some users it was not ready for a survey-based evaluation. The difficulties in simulating real conditions for the user context, and the composition of services based on the current user context, lead us to evaluate the performance of that components that present more challenges or even problems.

In our test scenario we used two computers connected to the campus wireless network (IEEE 802.11g).

Computer 1 (C1) is an Intel Core 2 Duo 7400 (2.4Ghz) 3GB DDR2 with OS X 10.5.5, and runs the iCas architecture middleware described in Section VI.

The Glassfish, that runs third party Web Services, is installed in computer 2 (C2), an Intel Core 2 Duo T8600 4GB DDR2 with Linux (kernel 2.6.24) as its operating system. Some of the third party services installed in this machine are services provided by the library, and e-learning platform.

B. *Context Engine Core test*

In this test we intended to get the first performance results from the following main components that are exposed to computationally and I/O intensive processes: context engine core (inserting data and querying for derived contexts), service composition and service execution. We excluded services discovery and selections because the selection is highly dependent on the context engine core.

Table 1 presents the results performed in C1. For each result three measures were made and the table shows the average time in milliseconds (ms) of theses measures.

The graphic in Fig. 9 shows the average time consumed by the Context Engine Core to execute one query, which saves context information into the persistent ontology database. It is possible to observe that the Context Engine performs well in terms of the data volume to store and the variation is gradual and linear. During these tests the persistent ontology database has reached 1GB in disk space.

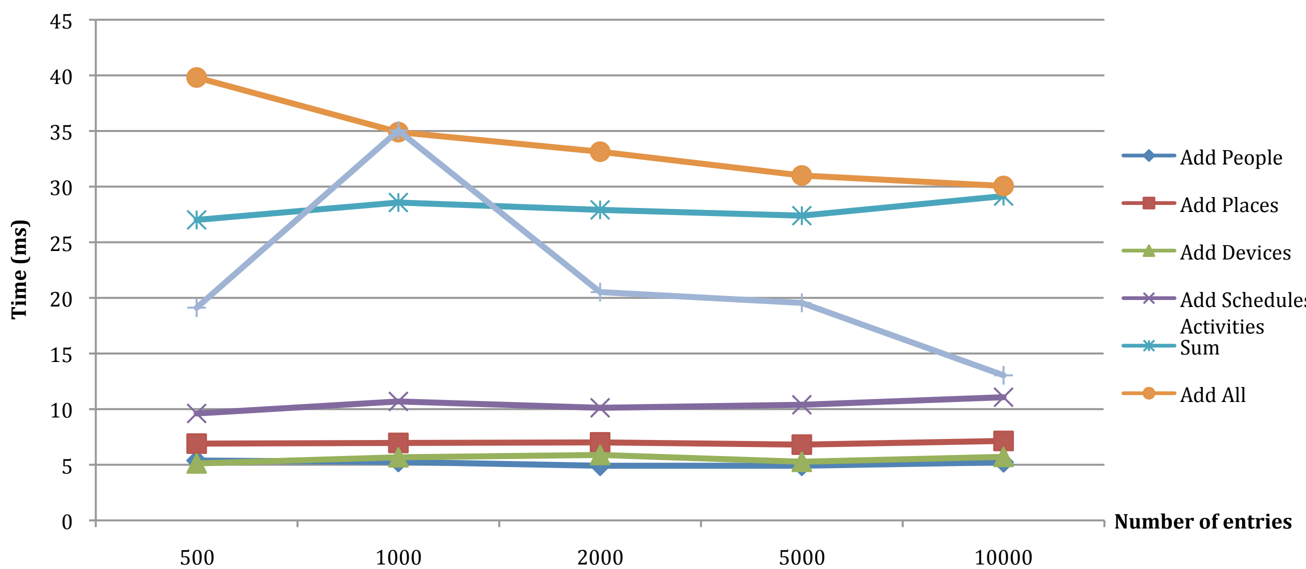


Figure 9 Times to insert context using the Context Engine Core.

It can be observed that the Context Engine Core is able to support intensive loads and that the use of persistent ontologies s not a problem, but it seems to be a good option. Nevertheless, this performance could be improved either by optimizing the DB engine parameters or by using a faster computer to host iCas and the database management system.

To test the reasoning component we executedtwo types of SPARQL queries:

- The first one was a simple query that returns the interests of a specific person and the time average to execute this query was 10ms.
- The second was a more complex query described in the section VI. This query, returns all the events related with a subject and where they are happening. The average time to execute this query was 80ms.

Finally an inference using the Pellet reasoner was executed to explore the resources of OWL language, more specifically the transitive property, already explained in Section VI. In this example the user location was inferred and the average time do to this operation was 304ms.

C. *Service Composition and Execution test*

Table 2 presents the results of testing the Service Selection Mechanism,described in Section VI. The second column shows the time to load the services descriptions and to check its consistency for different numbers of services, specified in column 1. It should be noted that this delay only occurs when iCas is initialized and the services are loaded, which not demand a quick response of this operation as occurs on the services selection process.

When a new service is added or removed to the services repository only that service ontology is added or unloaded, which is a fast operation. The third column shows the time consumed to select the services to deliver to the user.

Table 2 Times of the Selection Mechanism

Number of Services	Time to load and check the consistency (sec)	Services Selection (ms)
10	8,9	25,0
20	31,0	45,0
38	80,7	97,0

The graphic of the Fig. 10 shows the average time needed to load each service. Fig 11 illustrates the average time required to make a service selection.

In Fig. 10, it's possible to observe that the time required to select the services islow which enables to give a quick response to the users' requests.

Observing both figures, it's also possible to realise that the consumption of time per service, required for loading and checking it, and to the services selection, has a minimal increment as the number of services to use increases.

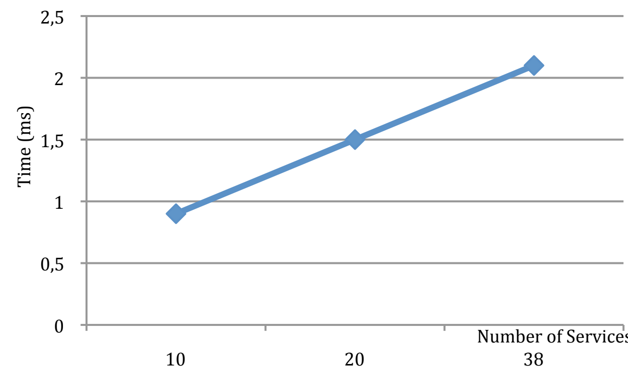


Figure 10 Load and check services process.

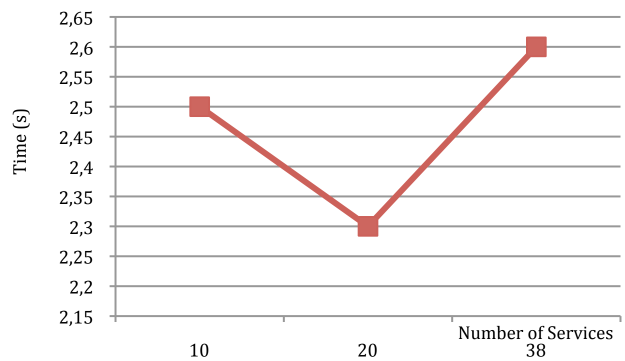


Figure 11 Service selection process.

To test the service composition and service execution we ran a client in C1, which launched a number of threads. Each thread intended to simulate a user that orders a service composition and its execution. Table 3 shows the test results of the Service Composition and Service Execution components.The test consisted in the variation of two parameters: the number of services used in a composition and the number of requests to perform the composition and its execution.

Each thread is responsible to make a unique request and to wait for the response.

The composition of services was the result of services joined in pipeline. The services that were part of this composition were provided by the application server running in the C2 machine, and had an execution time of 20ms. Our intention was to figure out how thesecomponents performed

with differentloads of service composition and execution. The maximum number of services used in a

Table 3 Times of Services Composition

	1 thread (ms)	100 threads (ms)	100 threads (time per thread (ms))	500 threads (ms)	500 threads (time per thread (ms))
2	287,8	26843	268,4	133058	266,1
4	410,5	27339	273,3	135900	271,8
8	528,8	28511	285,1	141471	282,9
10	642,0	30619	306,1		
12	780,0	33702	337,0		
14	820,0	X			
16	927,8	NT			

composition was 16, joint sequentially and the time consumedto execute this composition was 927ms.

To test limit conditions, we used this last composition(16 services)for a load of 100 requests and this component wasnot able to respond and it halted. Analyzing the time consumption of each thread to execute the composition of 12 services, it was 337ms, less than the time a unique thread took to execute the same composition (780ms). It's also possible to see that before the iCas frizzed the time to make and execute a service increased linear and gradually with the increase of the number of services used to make a composition.This problem willbe analyzed in future.

Using a composition of 8 services, this component was able to compose and executed requests made by 500 threads with the average time less than 300ms. In the future we also intend to test parallel compositions and the mix of pipelined and parallel workflow composition.

IX. Conclusion

In this paper we have presented iCas, a service-oriented architecture that uses an ontological context model to provide personal and contextual information and to support the composition of context-aware services. The two major contributions of our work are the joint use of a semantic context model (SeCoM), to describe and explore the expression of contextual information, along with the support of dynamic composition, of context-aware services by the user.

A prototype of the iCas platform has been implemented and functional tests have been conducted. Some experimental setups for services composition have been made using the iCas client prototype.

We also present the first performance evaluation in which we tested some of the main components of iCas, and found that the results of having a central server architecture to provide the had-hoc composition of services were encouraging.

A. *Limitations*

The current iCas implementation has some limitations. One is the granularity of services, i.e., which level of granularity the services should have to provide the best services to the user's needs. A fine-grained service addresses small units of functionality or exchange small amounts of data. Consequently, it will be more complicated to the user to build a service and to the architecture to orchestrate more services. Otherwise the coarse-grained services encapsulate more functionality reducing the number of services to make a composition, but they also hide the high level of functionality under one single interface and usually exchange more complex data, which might be harder to deal with.

Another problem is the transformation of standard web services into OWL-S services. There are tools to perform this task, but they have very limited functionality regarding service inputs, outputs and the range of these parameters, which are described by the service profile. If a service has complex datatypes (ex. structures, data collections), these tools are not able to perform that transformation. Some of these complex datatypes have to be described by the user, using the OWL and the service parameters can also be transformed using XSLT transformations, which are very susceptible to syntax errors.

For now, it is not possible to provide execution processing that depends on conditional statements, such as if-then-else and repeat-until, because they are not supported by the API. The API authors already announced the intention to include such functionalities in future versions.

Until now we have not tested the service composition in devices with limited resources, and the client prototype uses the standard Java Virtual Machine and Web Services.

There are also other limitations and challenges related with services compositions and the issues discussed in [28, 41], such as composition correctness, services dynamic availability and services trust,

B. *Future Work*

In the future we intend to finish the implementation of iCas and test it in a real scenario on a university campus. In this scenario we intend to determine how the context-aware mobile technologies can be used to assist pedagogical features and the socio-pedagogical interaction of various types of users.

References

[1] J. P. Sousa, E. Carrpatoso, and B. Fonseca, "A Service-Oriented Middleware for Composing Context Aware Mobile Services," in Internet and Web Applications and Services, International Conference on, Venice, Italy, 2009, pp. 357-362.

[2] R. Want, B. Schilit, N. Adams, R. Gold, K. Petersen, D. Goldberg, J. Ellis, and M. Weiser, "The Parctab Ubiquitous Computing Experiment," Mobile Computing, pp. 45-101, 1996.

[3] D. Salber, A. Dey, and G. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," 1999, pp. 434-441.

[4] D. Chakraborty, A. Joshi, T. Finin, and Y. Yeshadoi, "Service Composition for Mobile Environments," Mobile Networks and Applications, vol. 10, 2005.

[5] S. Panagiotakis and A. Alonistioti, "Context-Aware Composition of Mobile Services," IT Professional, vol. 08, pp. 38-43, 2006.

[6] T. Gu, H. Pung, and D. Zhang, "A service-oriented middleware for building context-aware services," Journal of Network and Computer Applications, vol. 28, pp. 1-18, /01// 2005.

[7] M. Sheshagir, N. Sade, and F. Gandon, "Using Semantic Web Services for Context-Aware Mobile Applications," in MobiSys 2004 Workshop on Context Awareness, Boston, 2004.

[8] L. Nan, Y. Junwei, L. Min, and S. Yang, "Towards Context-Aware Composition of Web Services," in Fifth International Conference on Grid and Cooperative Computing, Washington, DC, USA, 2006, pp. 494–499.

[9] W3C, "OWL-S: Semantic Markup for Web Services," 2004.

[10] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in Mobile Computing Systems and Applications, 1994. Proceedings., Workshop on, 1994, pp. 85-90.

[11] P. J. Brown, "The Stick-e Document: a Framework for Creating Context-aware Applications," in Proceedings of EP'96, Palo Alto, 1996, pp. 259-272.

[12] N. S. Ryan, J. Pascoe, and D. R. Morse, "Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant," in Computer Applications in Archaeology, Oxford, 1998.

[13] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, London, UK, 1999, pp. 304-307.

[14] H. Lieberman and T. Selker, "Out of context: computer systems that adapt to, and learn from, context," IBM Syst. J., vol. 39, pp. 617-632, 2000.

[15] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: standard ontology for ubiquitous and pervasive applications," in Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on, 2004, pp. 258–267.

[16] R. F. Bulcão Neto and M. G. C. Pimentel, "Toward a Domain-Independent Semantic Model for Context-Aware Computing," in 3rd Latin American Web Congress (LA-Web'05), Argentina, 2005, pp. 61-70.

[17] T. R. Gruber, "A translation approach to portable ontology specifications," Knowl. Acquis. %@ 1042-8143, vol. 5, pp. 199-220, 1993.

[18] R. F. Bulcão Neto and M. G. C. Pimentel, "Performance evaluation of inference services for ubiquitous computing," in XII Braziliam Symposium on Multimedia and Web Systems, Brazil, 2006, pp. 27-34.

[19] G. D. Abowd, E. D. Mynatt, and T. Rodden, "The human experience," IEEE Pervasive Computing, vol. 1, pp. 48-57, 2002.

[20] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, The Description logic handbook: Theory, implementation, and applications: Cambridge University Press, 2003.

[21] D. Brickley and R. V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema," W3C, 2004.

[22] G. Schreiber and M. Dean, "OWL: Web Ontology Language Reference. W3C Recommendation," 2004.

[23] R. F. Bulcão Neto, A. A. Macedo, J. A. Camacho-Guerrero, and M. G. C. Pimentel, "Configurable semantic services leveraging applications context-aware," in Proceedings of the 11th Brazilian Symposium on Multimedia and the web, Brazil, 2005, pp. 1-9.

[24] J. Allen, "Maintaining knowledge about temporal intervals," Communications of the ACM, vol. 26, pp. 832–843, 1983.

[25] G. Alonso, F. Casati, H. Kuno, and V. Machiraju, "Web Services - Concepts, Architectures and Applications," 2003.

[26] H. K. Cheng, Q. C. Tang, and J. L. Zhao, "Web Services and Service-Oriented Application Provisioning: An Analytical Study of Application Service Strategies," Engineering Management, IEEE Transactions on, vol. 53, pp. 520-533, 2006.

[27] M. P. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on, pp. 3-12, 2003.

[28] N. Milanovic and M. Malek, "Current Solutions for Web Service Composition," IEEE Internet Computing, vol. 8, pp. 51-59, 2004.

[29] Oasis, "UDDI v3.0 Ratified as OASIS Standard," 2005.

[30] A. Ankolekar, "DAML-S: Web Service Description for the Semantic Web," 2002.

[31] J. Yang and M. Papazoglou, "Web Component: A Substrate for Web Service Reuse and Composition," in CAiSE '02: Proceedings of the 14th International Conference on Advanced Information Systems Engineering, London, UK, 2002, pp. 21-36.

[32] R. Milner, F. L. Bauer, W. Brauer, and H. Schwichtenberg, "The polyadic pi-calculus: a tutorial," in Logic and Algebra of Specification: Springer-Verlag, 1993, pp. 203-246.

[33] R. Hamadi and B. Benatallah, "A Petri net-based model for web service composition," in ADC '03: Proceedings of the fourteenth Australasian database conference, Darlinghurst, Australia, Australia, 2003, pp. 191-200.

[34] T. Bultan, X. Fu, R. Hull, and J. Su, "Conversation specification: a new approach to design and analysis of e-service composition," in WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, 2003, pp. 403-410.

[35] B. Srivastava and J. Koehler, "Web service composition - current solutions and open problems," in ICAPS 2003 Workshop on Planning for Web Services, 2003.

[36] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1. W3C Note," World Wide Web Consortium (W3C), 2001.

[37] UPnP.org, "UPnP Forum - Standards," in Standards, 2009.

[38] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDF," W3C, 2008.

[39] W3C, "XHTML™ Modularization 1.1, W3C Proposed Recommendation," 2008.

[40] E. Sirin, J. Hendler, and B. Parsia, "Semi-automatic Composition of Web Services using Semantic Descriptions," in Proc of Web Services: Modeling; Architecture and infrastructure workshop in conjunction with ICEIS2003, 2003.

[41] M. Bourimi, F. Kühnel, and D. e. D. I. Abou-Tai, "Tailoring collaboration according privacy needs in real-identity collaborative systems," in CRIWG 2009 - 15th Collaboration Researchers' International Workshop on Groupware Peso da Régua, Douro, Portugal, 2009.

International Journal on Advances in Software, vol 2 no 2&3, year 2009, http://www.iariajournals.org/software/

A Model and an Implementation Approach for Event-Driven Service Orientation

Olga Levina, Vladimir Stantchev
SOA and Public Services Research Group
Berlin Institute of Technology
10578 Berlin, Germany
Email: olga.levina|vladimir.stantchev@tu-berlin.de

***Abstract*—Event-driven architecture is gaining momentum in research and application areas as it promises enhanced responsiveness, flexibility and advanced integration. The combination of event-driven and service-oriented architectural paradigms and web service technologies provide a viable possibility to achieve these promises. This article is an extended version of an ICIW 2009 conference paper and introduces several aspects that can facilitate such combination. It presents an event model, outlines an architectural design and proposes sample implementation technologies. The ongoing evaluation in real-world scenarios confirms the applicability of the approach for the realization of web services-based event-driven architecture.**

***Keywords*-web services; event-driven architecture; service-oriented architecture; business events; business rules**

I. Introduction

Physical systems supporting business processes are increasingly coping with the effects of external changes and inputs. This information is used to monitor and control the process flow but it also creates new requirements for underlying network and application system structure. Asynchronous and data-centric communication in a distributed system is an approach followed by designers promoting event-driven and service-oriented architectures. Ubiquity and functional independence are some of the value adding characteristics of Service-Oriented Architecture (SOA). Asynchronous communication, interest-based message delivery using the publish/subscribe principle and event orientation by providing event sensors and event processing components are the characteristics of an Event-Driven Architecture (EDA). This article is an extended version of our ICIW 2009 conference paper [1] and motivates the implementation of a holistic architecture: Event-driven service-oriented architecture (ED-SOA) for combing function- and data-centric views on IT systems and enterprise as a whole. The combination of the two approaches is an actively discussed topic among information systems researchers, IT architects and vendors. This paper provides needed definitions and structures to promote common understandings and terms. Furthermore, reference architecture of an ED-SOA is proposed. Web services are suggested as the realization technology. This decision is confronted with the ongoing research and development results for enterprise event-driven systems.

The remainder of this article is organized as follows: in Section II we provide the definitions of EDA, SOA and web services. We introduce a reference architecture of an ED-SOA in Section III and present a realization approach based on web services and Quality-of-Service (QoS) assurance (Section IV). Related work on technology for the implementation of enterprise event-driven systems is provided in Section V. Discussion of our approach and outlook to future working areas complete the article.

II. Definitions

This section introduces some definitions that we use throughout the article.

A. Service-Oriented Architecture

Service-oriented architecture is one of the most discussed topics in the IT these days. Since there is no common SOA definition yet, the term is used as a combination of elements of software architecture and enterprise architecture. It is based on the interaction with autonomous and interoperable services that offer reusable business functionality via standardized interfaces. Services can exist on all layers of an application system (business process, presentation, business logic, data management). They may be composed of services from lower layers, wrap parts of legacy application systems or be implemented from scratch [2]. Service-orientation as a design paradigm roots in several already known approaches such as object-orientation, aspect-oriented programming (AOP), enterprise application integration (EAI) and business process management (BPM) [3]. Following service-orientation approach a system is decomposed in its functionalities. A service is hence an element that encapsulates a business function and cannot be further decomposed without harming its functionality. Services can be defined as autonomous, platform-independent entities that can be described, published, discovered and assembled [4]; they are technologically neutral, loosely coupled and support location transparency encapsulating business functionality [5]. There are different ways to implement distributed services into IT architecture. They can be implemented using data-based [6], object-oriented (e.g. CORBA and Java RMI) or service-oriented approaches. Since the data-oriented approach applies only to structured data [6] and object-oriented approaches do not necessarily enable loose coupling and ubiquitous services access [7], service implementation today is often done using web services. Service orientation and SOA can be used best, when processes or their parts are standardized, when they are often repeated without changes, or when multiple users need the same process component to complete their tasks. Service invocation (consumption) in an SOA is realized remotely using RPC-like procedure and on request of the service consumer. This approach allows an explicit request for a WSDL-defined service interface to be invoked using SOAP message exchange.

B. Event-Driven Architecture

An event-driven architecture is a structure in which elements are triggered by events. An event in the enterprise context is a change in the state of one of the business process elements that influences the process outcome. Being abstract constructions, events are captured as event objects. An event object allows a machine to process, calculate and manipulate the event. Main components of an EDA are: event sources or generators, event recipients or consumers, event sensors and event processors. Event source(s) and event consumers are connected either directly (point-to-point) or via a middleware or broker (bus). Event source might be an application, business process, internal or external stakeholder or any other abstract data change [8]. Event recipients are all interested subscribers. Event capturing and delivery must be guaranteed by compatibility standards and can be processed in an extra component – the event agent. The logic of collecting and routing of events is captured in the event processor. Incoming event(s) are processed and forwarded to event consumers in (predefined and "soft") real-time. An event consumer reacts to received events by performing its functionality or publishing an alert. There are three types of events that need to be processed: single event, event stream and complex event(s). The difference between an event stream and a complex event can be described as event stream being a temporal sequence of event objects in the "first come-first-serve" manner [9] and complex events being a group of events that contains elements from different contexts or different time points. Processing events means performing operations on event objects like creating, transforming, reading or deleting. Algorithms for processing of multiple or interlaced events are summarized in complex event processing (CEP) technique. It allows identification and extraction of structured information from message-based systems. CEP includes event analysis and correlation delivering a decision triggering information. CEP uses business rules as well as patterns, maps and filters to specify relationship between events [10]. Event monitoring is facilitated by business activity monitoring (BAM) tools. These tools are often a part of a business process management suite and are currently more focused on detecting events and visualizing them on a dashboard than on automated decision making, therefore requiring less computational intelligence. Event-driven systems provide real-time visibility of the observed processes and allow almost real-time reaction.

In this article we show that a SOA can provide suitable conceptual structure for an EDA. Contrary to communication in SOA, EDA components interact asynchronously, event processor being a connector with high intelligence. In EDA event sources and event recipients do not know anything about each other, neither does event source know whether and what kind of reaction was caused by its appearance. Figure 1 shows an exemlary EDA architecture.

For further event processing and capturing during the requirements analysis or modeling phase, an event structure is needed. Figure 2 shows our proposed event model that allows a distinct description of any generic event. In this article we focus on business events, i.d. state changes of a business entity. This definition differs from the one in the context of event-driven distributed information systems like CORBA, where an event is defined as the occurrence of some interaction point between two computational objects in a system [11]. This kind of event or event description languages will not be considered for modeling, since state changes of business objects are our primary concern.

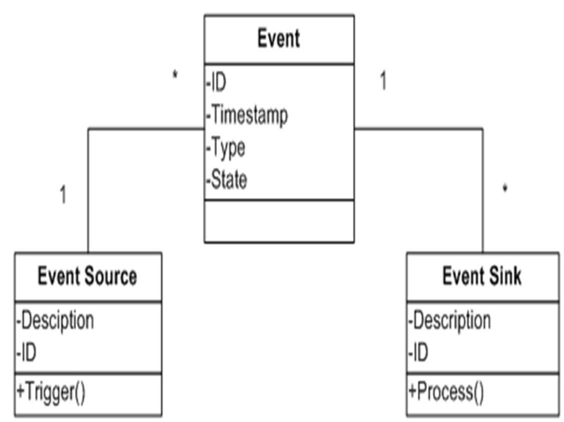


Figure 2. Structured Event Model

The event structure shown in Figure 2 shows the main components involved into event creation and processing. In the context of an event-driven architecture there are system elements that act according to the changes in states of other objects. That means that the event sources are being observed by the event sinks considering their change of states. The event source is described by its unique ID and a description, e.g., the name of the source, in natural language. Possible operation that can be performed by the source is triggering the event when the change of the state occurs. Here the main assumption is that an event, i.e. a state change, can originate only from one source. An event is identified by a unique ID, timestamp, event type and the current state of the event source. Timestamp is needed to compute the time

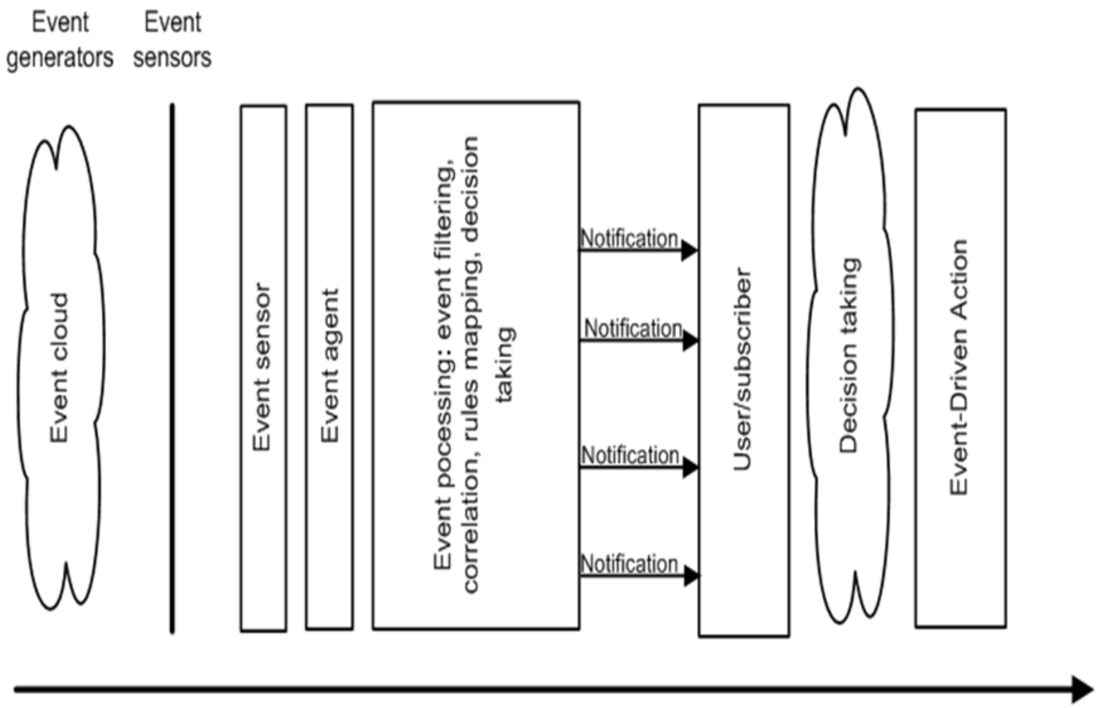


Figure 1. Event-Driven Architecture

model of the incoming events and to provide information for composition of complex events. Event type corresponds to the event type mentioned above: single event, complex event or event stream. This information can be processed by the sink in order to react to the incoming data. Event sink is an architectural object or element that is interested in the state change of the event source it is subscribed for, i.e. its actions are triggered by the state changes of the event source. Event sink description in natural language as well as an ID are used as its attributes. Event sink can also provide the first processing step of the incoming events. These processing can include queuing the events that are part of the event stream or combining the events to a complex event.

Our event model consists of the following formal elements:

- S is the set of event sources included in the model.
- S_n is the set of event sinks included in the model.
- Z is the set of the object states, while Z_s is the set of the object states Z of the source s.
- E is the set of events considered in the model.
- T is the set of possible event types, with: $T = single, complex, stream.$
- TS is the set of the timestamps, with: $TS = day, month, year, hour, minute, second.$

These aspects can be captured and modeled using a modeling eclipse Plug In, called Visual Event, Figure 3 shows the stand-alone event including event source, sink, and the event itself including its attributes and the data types of the attributes. It is also possible to comment on the model elements.

Using Visual Event Plug-in, it is possible to model all the events that are needed to trigger an action of the event sink using the annotation at the control flow. Additional information spaces are included in the diagram properties to take account of the sequence number of the event, timestamp, data type, etc. when modeling event sinks and sources. The Visual Event plug-in is comprised of an event, with event name, attributes and description, and an event trigger. The event trigger is the source that changes its states and thereby triggers an event. Event sinks are subscribers for a specific event occurrence in a publish/subscribe implementation paradigm.

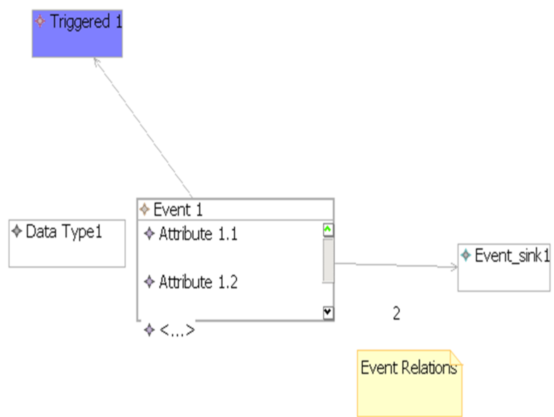


Figure 3. Event Modeling Plug-In

III. Combining EDA and SOA

Both SOA and EDA have characteristics that complement each other. Both use services but differ in the way they addressed a service to be invoked. SOA provides loosely coupled techniques like web services but its functionality is tightly coupled to the Request/Respond mechanism while EDA provides an asynchronous communication and loose coupling [12].

While SOA offers EDA a suitable design approach by providing a distributed environment for separating business logic, processes and technical functions, it benefits from another service invocation technique that loosens the rigor of the RPC-style calls. When observing these characteristics the merged structure of the three concepts provides, one can realize multiple synergy aspects. Service-orientation allows to capture and store events as services. Integration of legacy systems into service-oriented architecture may be done using the derived business rules the systems are using, or by using event-driven architecture. SOA is based on a remote access principle allowing a distributed environment, necessary for both event-driven architecture and business rules. EDA has a decoupled, asynchronous structure that complements loose coupling and synchronous communication of SOA [13]. Implementing SOA-suitable environment means implementing an environment where events can operate on their best and many architectural interactions are already standardized. Further synergies come up with communication and process management in a distributed system, which can be assured by adopting a business rules oriented ED-SOA. Often having a highly distributed architecture, enterprises create benefits from the real-time information availability. EDA provides a structure that allows a fast reorganization of business processes without affecting application or technical structures. Fast reaction to environmental changes in is possible without the need to adapt technical infrastructure. Functional decomposition on a high-granularity level, that is crucial for robustness to change of a system, is provided by SOA. Merging these concepts results in an enterprise architecture that is more flexible while being robust to changes. Its components are loosely coupled and can be accessed in any business situation.

The major aim of enterprise architecture is realized in the ED-SOA concept by SOA combining business functions and IT, and EDA focusing on data as well as business relevant event orientation; both SOA and EDA concepts can be used for application and legacy systems integration [14]. Covering the aim and component spectrum of enterprise architecture as described above, ED-SOA can be regarded as its evolution. Figure 4 shows a proposed ED-SOA reference architecture including security aspects, business rules processing and business data integration. Components that can be encapsulated as services are named. They were identified according to the main principles of service-orientation: their granularity is can be easily identified and discovered while being reusable by different components in different points of time. The concrete integration infrastructure into the application systems landscape depends on the technology used to realize ED-SOA. Here an enterprise service bus (ESB) is a suitable solution as the architecture is to be realized using web services.

IV. Enabling ED-SOA

After modeling the event and defining its specific structure, it can be realized technically using web service technology. Web services are currently the most promising service-oriented technology [15]. They use the Internet as the communication medium and open Internet-based standards, including the Simple Object Access Protocol (SOAP) for transmitting data, the Web Services Description Language (WSDL) for defining services, and the Business Process Execution Language for Web Services (BPEL4WS) for orchestrating services.

The Visual Event diagram (Figure 2)also delivers a XML-structure. Event content and its processing components such as sinks and sources can be derived from the event model as shown in figure 3 and implemented as a event service in a service-oriented architecture. This approach, first defining and modeling events for their further implementation using Web Services, allows a structured way to design and manage EDA conserving its main principle of agility and loose coupling. Modeling plug-in developed and presented here supports the easy implementation generating a XML-code of the event content. SOA provides important standards and tools, like WSDL and UDDI, for describing, storing and finding of the events within the architecture.

This section provides an overview of implementation technologies that we used in our proof-of-concept and is structured according to the elements presented in Figure 4.

Software components that call (consume) services can be developed in a variety of languages on a variety of platforms.

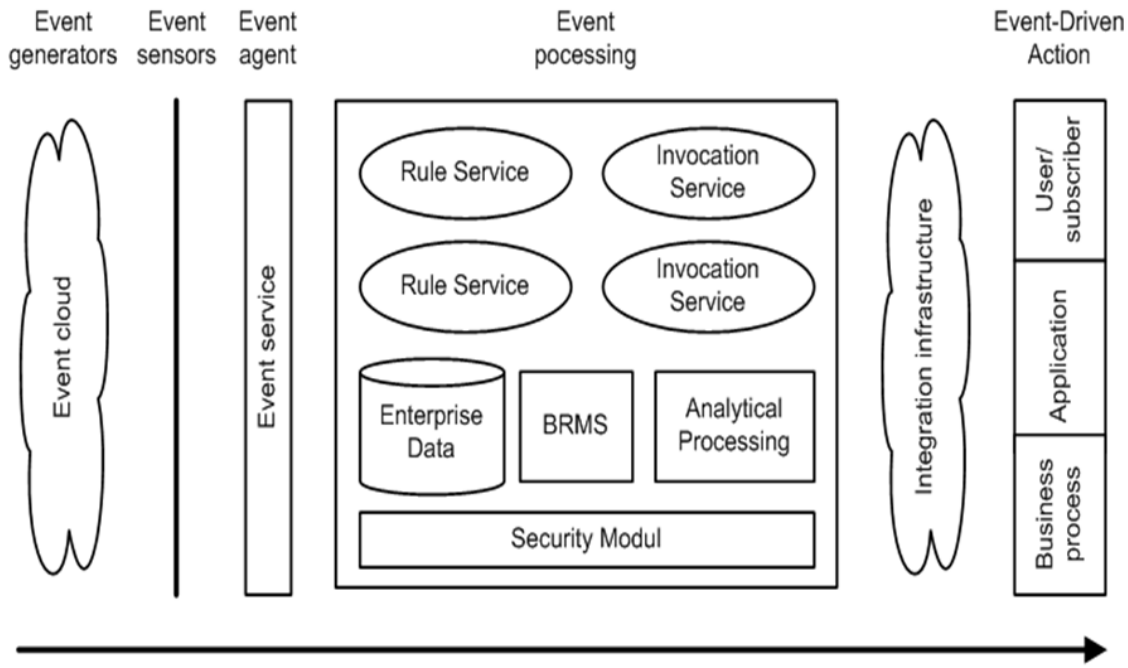


Figure 4. Event-Driven Service-Oriented Architecture

Typical integrated development environments (IDEs) allow this interaction without the need to code SOAP messages. They generate a proxy stub object on the local machine that marshals calls to the actual web service. Therefore, from a software engineering point of view a single service interaction is not much different from the interaction of COM (Component Object Model) or CORBA components. Important new aspect of web services is the promise of automatic composition going beyond the binary integration of COM and CORBA. Such flexible processing infrastructure can adapt more easily to changes in the functional requirements of an event-driven business process.

A. Platforms

The complexity involved in providing a single web service is often underestimated. A look at hardware platforms, even commodity hardware, reveals complex microprocessors and processing architecture. Standard OSs are far away from microkernel designs [16] such as Mach [17] and contain a large number of OS extensions. These are called *modules* in a Linux system [18] and *drivers* in a Windows system. [19]. Beside typical device drivers, extensions include network protocol implementations, file systems and virus detectors. Extensions are more than 70% of the Linux source code [20], while Windows XP includes over 35,000 drivers with over 120,000 versions [21]. Typical component frameworks such as .NET and J2EE often serve as the middleware for providing web services [22]. Therefore, we selected the .NET Framework as platform. A more detailed look at the application programming interfaces of these environments [23] and [24] reveals their complexity.

B. Quality of Service and Nonfunctional Properties

The nonfunctional properties (NFPs) of a software system are those properties that do not describe or influence the principal task / functionality of the software, but are expected and can be observed by end users in its runtime behavior [25].

QoS encompasses important NFPs such as performance metrics (for example, response time), security attributes, transactional integrity, reliability, scalability, and availability. Traditionally, QoS is a metric that quantifies the degree to which applications, systems, networks, and other IT infrastructure support availability of services at a required performance level [4]. Web services environments are based on flexible composition of services and therefore demand greater availability of applications. Furthermore, they introduce increased complexity in terms of delivering, accessing and managing services.

The existing standards for specification of QoS characteristics in a service-oriented environment can be grouped according to their main focus: software design/process description (e.g. UML Profile for QoS and QML - QoS Modeling Language [26], service/component description (e.g. WS-Policy) and SLA-centric approaches (e.g. WSLA - Web Service Level Agreements [27], WSOL - Web Service Offerings Language [28], SLAng - Service Level Agreement definition language [29] and WS-Agreement [30].

Extensive research concerning NFPs also exists in the field of CORBA (Common Object Request Broker Architecture), particularly in the areas of real-time support [31], [32], replication as approach for dependability [33], [34], [35], [36], adaptivity and reflection [37], [38], as well as mobility [39], [40].

The approach we apply to formalize and control NFPs is called architectural translucency [41] – the ability to consider reconfiguration options at different system levels and understand their effects on the performance-related NFPs of a system. It allows us to specify service level objectives [42] and to enforce them by replication at different architectural levels, e.g., operating system [43] or service framework [44].

C. Implementing Rule and Decision Services

Our sample implementation uses the .NET Framework as a serviceware and the Microsoft Workflow Foundation (included in .NET 3.0) as basis for the rule and decision services. The workflow foundation supports different types of workflows (see Figure 5) and facilitates particularly the implementation of rules-based activities. Using it, we can map rules defined at the business level to any .NET programming language in a straightforward way.

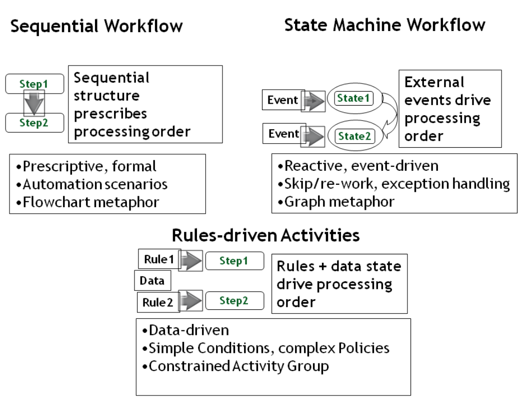


Figure 5. Support for Rules and Events in Microsoft Workflow Foundation (Source: Microsoft)

D. Implementing Invocation and Notification Services

Any step in our workflow (as implemented in the Microsoft Workflow Foundation) can call operations on other objects on the same machine, invoke other workflows or directly invoke web services. Events that trigger a state change (next step) of a workflow range from sensor information (e.g., RFID) through changes in data sources (e.g., relational databases) to web service outputs or fault messages. There are several integration depths that we regard as relevant for events:

- Events at the data level – here we differentiate between events originating from database management systems (DBMS), e.g., relational databases, and events originating from sensors, e.g., RFID readers and scanners.
- Events at the object level – these are typically state changes in class instances which we regard only if they are manifested by public methods.
- Events at the service level – a call, or a response of a service.

Generally, we can map the "lower level" events (data and object) to the service level using web service wrappers. Furthermore, we can combine events to complex events (e.g., a delivery has arrived and a warehouse is full) by using composite services. In the context of this composition we particularly regard the NFPs of the composed service, as described in [25].

E. Integration Aspects

An already agreed-upon SOA strategy greatly facilitates our approach as we can then expect that critical software functionality will be provided as web services in the specified timeframe. If our approach has to be integrated in more heterogeneous environments we can benefit from the capabilities of .NET 3.0 to interact with diverse remote components, such as other .NET objects, SQL servers and web technologies.

F. Application Scenarios

One application scenario that can greatly benefit from ED-SOA is logistics. Our demo application in this domain (more particularly contract logistics) differentiates between several states of a shipment that is being transported (see Figure 6). It begins with an initial event (*Container sent*) and goes through the following statuses: *Fetched, Accepted, Loaded, Unloaded, in Delivery,* and *Delivered*. Business users can define rules related to these statuses and corresponding events (e.g., a longer delay or a missed deadline) using a web-based user interface. We then use this rule specifications in our implementation to trigger next (or additional) steps in the workflow according to incoming events. Events can be propagated in a variety of ways: RFID-based communication in a warehouse system, e-mail notifications, changes in inventory databases, as well as other components or web service calls and responses. This makes our approach highly flexible to changing business requirements – they can be submitted to our system as a new rule set via the user interface. Figure 7 shows an overview of our architecture on a given site (e.g., intermediate warehouse or point of delivery). It integrates an event processing component, components for event sensing, a supply-chain-management system, as well as to engines – the rules engine and the architectural translucency (AT) engine. The AT engine is responsible for service level enforcement with respect to NFPs.

One other specific domain that can greatly benefit from ED-SOA is healthcare. We have applied our approach in the area of clinical processes and their optimization based on localization techniques [45]. The requirements of the scenario – localization of a large number of mobile devices (10,000) within a refreshing interval of five seconds, make the architectural integration a challenging task. The scenario is described in details in [45], aspects of the service-oriented integration and service level assurance in [46].

G. System Evaluation

We conducted our evaluation twofold – using empirical evaluation methods as well as system-oriented performance evaluation. Our empirical evaluation follows the usablity evaluation methodology presented in [47]. In our healthcare scenario we used questionnaires and expert interviews as usability evaluation test methods. These were addressed at clinicians that use our system. An overview of the surveyed group is given in Table I, a summary of results is presented in Table II. Overall, there is a substantial ($\Delta > 50\%$) increase in usability.

To ensure QoS aspects in our application scenarios we apply the method of architectural translucency [41], [46]. In this article we present excerpts from the performance evaluation at one site of our logistics scenario. The specific replication configuration used is shown in Table III.

The results of our performance evaluation are shown in Table IV. We anonymized the names of the web services due to non-disclosure requirements.

Test results show that replication at the OS level improves performance by approx. 25%, while replication at the serviceware level leads to improvement by 5-7%. Dual replication led to improvement by 7-15%.

1) Confidence Intervals: Results in Table IV are average results from six consecutive test runs. Each test run included tests of every replication setting for 120 minutes with 1 second think time before a request. This corresponds to some 7200 requests that were sent to each setting.

All tests for Web Service 1 resulted in 7200 requests served for all replication settings, so here the confidence interval is clearly 100%. All other confidence intervals are between 99% and 100%.

V. Related Work

Distributed event processing and event-driven systems became popular in recent years as the technology needed to provide and support these systems is rapidly evolving. In the 1980s and 1990s message-oriented middleware was used to facilitate integration of various application systems within an enterprise. Basic event-processing can be regularized by inclusion of Java Message Service and message-driven beans in Java Enterprise Edition (J2EE) [48]. Message-oriented middleware allows a push-based, publish-subscribe data-centric communication through message brokers or queued messages. As for the embedded, real-time systems based on event-orientation, they are often written in languages such as C or C++, with real-time services provided by CORBA (Common Object Request Broker Architecture) [48], [31]. CORBA also provides a publish-subscribe mechanism by the CORBA/IIOP (Internet Inter-ORB Protocol) [49].

Composition of applications from web services is governed by different requirements than typical component-based software development and integration of binary components. Application developers and users do not have access to documentation, code or binary component. Instead, they rely only on a rudimentary functional description offered by WSDL. Services execute in different contexts and containers, they are often separated by firewalls and can be located practically everywhere. This leads to a set of specific requirements a composition mechanism must satisfy as identified in [50]: connectivity, NFPs, correctness, automatic composition and scalability.

Every composition approach must guarantee connectivity. With reliable connectivity, we can determine which services are composed and reason about the input and output messages. However, since web services are based on message passing, NFPs, such as timeliness, availability, and performance must also be addressed. Correctness of composition means that the NFPs of the composed service must be verified. Automatic composition is the ability to automatically perform goal-based composition. Finally, composition of services within SOA must scale with the growth of business services that are based on composed technical services.

With the native capabilities of web services fully developed, several approaches for service composition started to emerge. The first generation composition languages were Web Service Flow Language (WSFL), developed by IBM, and Web Services Choreography Interface (WSCI), developed by BEA Systems. However, these proposals were not compatible with each other, and this led to the development of second generation languages. The most popular of them is BPEL4WS [51], which is a joint effort of IBM, Microsoft, SAP, Siebel and BEA. It originates in the combination of first generation languages (WSFL and WSCI) with Microsoft's XLANG specification.

SWORD is an approach, together with a tool set, for rule-based service composition. Here a service is represented by a rule that expresses that given certain inputs, the service is capable of producing particular outputs [52]. A rule-

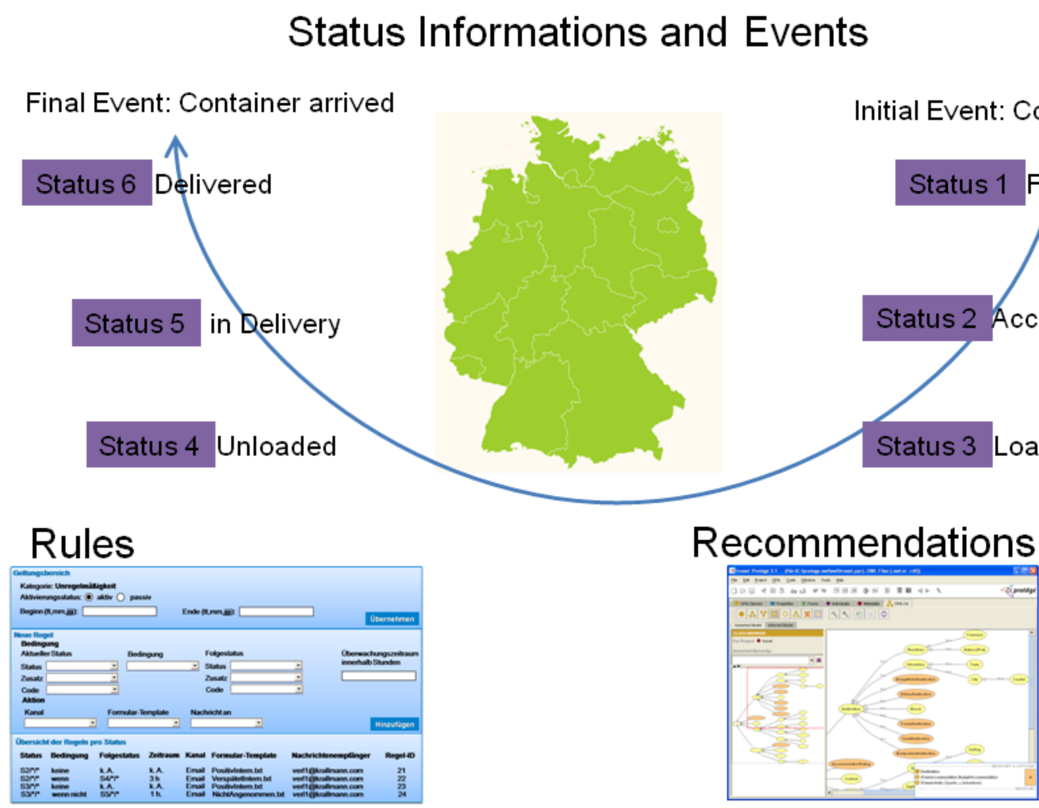


Figure 6. Application Scenario in Contract Logistics

Group Characteristics	No. of Participants	Percentage
Nurses	10	33.33
Surgeons	6	20
Anesthetists	3	10
Management	1	3.33
Other	10	33.33
Total	30	100

Table I
PROFILE OF TARGET GROUP FOR THE EMPIRICAL EVALUATION OF OUR HEALTHCARE SCENARIO

based expert system is then used to automatically determine whether a desired composite service can be realized using existing services. If so, this derivation is used to construct a plan that when executed instantiates the composite service.

Typically, SWORD does not require wider deployment of emerging service-description standards such as WSDL and SOAP.

Authors claim that although SWORD's expressive capabilities are weaker, the abstractions it exposes capture more appropriately the limited kinds of queries supported by typical web services which leads to simplicity and higher efficiency.

EFlow [53] is a platform for specification, composition and management of composite services. It uses a static method for workflow generation. Hereby a composite service is modeled by a graph that defines execution order of participating nodes. Graph creation is done manually, but subsequent graph updates can be automated. A graph may include service, decision and event nodes. Service nodes represent the invocation of atomic or composite services. Decision nodes specify workflow alternatives and decision rules. Event nodes allow services to send and receive certain types of events. Graph arcs show the execution dependency among nodes.

VI. DISCUSSION AND OUTLOOK

In this article we introduced the concept of an event-driven service-oriented architecture (ED-SOA) and proposed several aspects for its realization, such as an event model and a reference structure. Furthermore, we provided technology

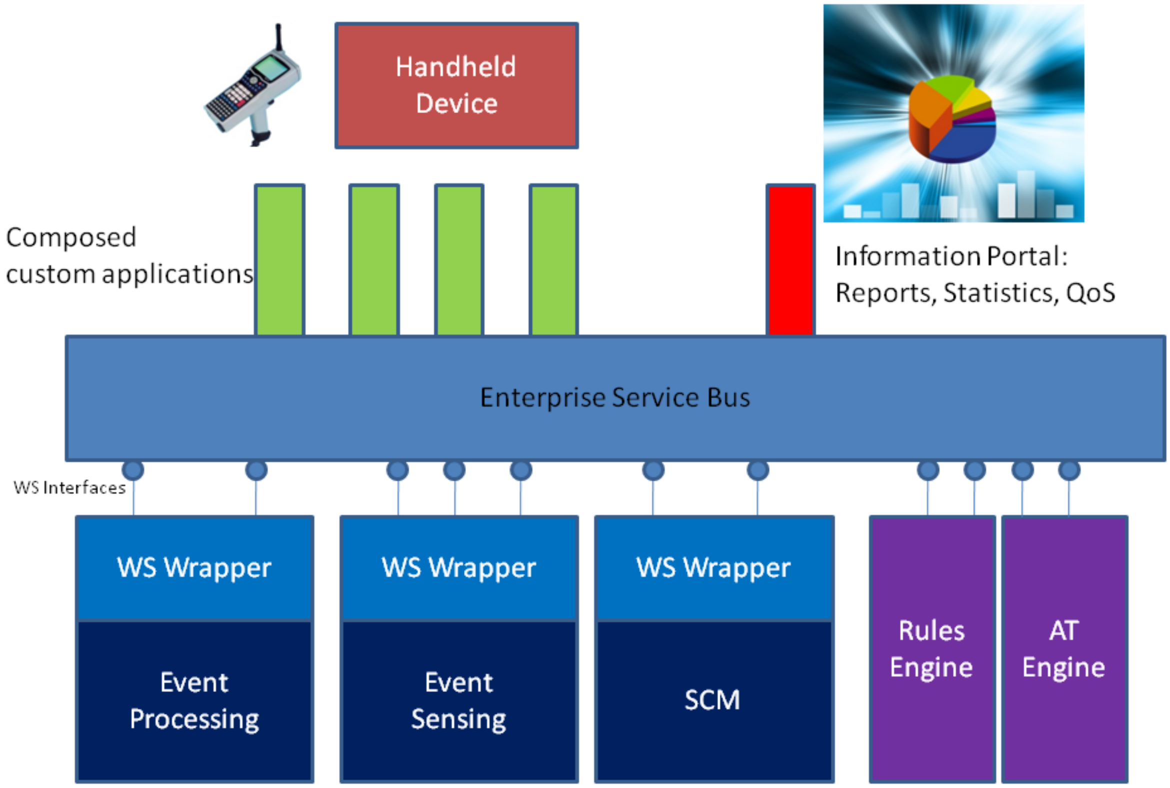


Figure 7. General Site Architecture for ED-SOA Implementation in Logistics: SCM denotes the Supply-Chain-Management System, AT Engine denotes the architectural translucency engine

Dependent Variables	**before**	**after**
Average Patient Preparation Time (min.)	31.20	16.30
Avg. Additional Preparation Tasks Needed (Nr.)	12	7
Avg. Number of Process Errors (perioperative)	6	2
Avg. Number of Process Errors (postoperative)	5	2
Clinicians: User Satisfaction (percent)	46	83
Patients: User Satisfaction (percent)	52	89

Table II
SUMMARY OF USABILITY EVALUATION RESULTS

and element definitions and outlined possible advantages of combining service-oriented and event-driven approaches for which we proposed a reference architecture. We regard our holistic approach as an important contribution that builds on many related concepts currently under development in this area.

The article also presented two application scenarios. Our application scenario in contract logistics used web services and the .NET Framework as enabling technologies and demonstrated major benefits of the approach. The empirical evaluation of our approach demonstrated increased user satisfaction, while its performance evaluation provided results that show its applicability for the assurance of QoS aspects within ED-SOA. Our future work lies in the areas of incorporation of predefined rule sets for specific domains (e.g., environmental conservation, privacy and security, healthcare applications) in the approach. This will allow us to provide a generic rule set that can be customized and extended according to the specific user requirements. The customization will be supported by a reference process for projects we are currently designing. We are also working on the further integration of various high-assurance techniques [41], [46] into the approach.

Setting:	Replication OS	Replication S
Setting 1: No Replication	n	n
Setting 2: Replication at OS	y	n
Setting 3: Replication at S	n	y
Setting 4: Dual Replication	y	y

Table III
REPLICATION SETTINGS FOR QOS ASSURANCE

Setting / Call	Web Service 1	Web Service 2	Web Service 3	Web Service 4
Setting 1:	7200	4176	4245	3578
Setting 2:	7200	5342	5418	4120
Setting 3:	7200	4424	4312	3692
Setting 4:	7200	4861	4803	3711

Table IV
TEST RESULTS AT DIFFERENT ARCHITECTURAL LEVELS - .NET AND WINDOWS ENVIRONMENT.

REFERENCES

[1] O. Levina and V. Stantchev, "Realizing Event-Driven SOA," in *ICIW '09: Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*. Los Alamitos, CA, USA: IEEE Computer Society, May 2009, pp. 37–42.

[2] H. Krallmann, C. Schröpfer, V. Stantchev, and P. Offermann, "Enabling autonomous self-optimization in service-oriented systems," in *Proceedings of The 8th International Workshop on Autonomous Systems - Self Organisation, Management and Control*. Berlin, New York: Springer, 10 2008, pp. 127–134.

[3] T. Erl, *Soa: principles of service design*. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.

[4] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-oriented computing: State of the art and research challenges," *Computer*, vol. 40, no. 11, pp. 38–45, Nov. 2007.

[5] M. P. Papazoglou, "Service-oriented computing: concepts, characteristics and directions," *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*, pp. 3–12, 2003.

[6] C. Batini, M. Lenzerini, and S. B. Navathe, "A comparative analysis of methodologies for database schema integration," *ACM Comput. Surv.*, vol. 18, no. 4, pp. 323–364, 1986.

[7] Y. Baghdadi, "A business model for deploying web services:a data-centric approach based on factual dependencies," *Information Systems and E-Business Management*, vol. 3, no. 2, pp. 151–173, 2005.

[8] H. Herbst, G. Knolmayer, T. Myrach, and M. Schlesinger, "The specification of business rules: A comparison of selected methodologies," in *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle*. New York, NY, USA: Elsevier Science Inc., 1994, pp. 29–46.

[9] B. Michelson, "Event-driven architecture overview- event-driven soa is just part of the eda story," Patricia Seybold Group, Tech. Rep., 2006.

[10] D. C. Luckham and B. Frasca, "Complex event processing in distributed systems," Stanford University, Tech. Rep., 1998.

[11] C. Ma and J. Bacon, "Cobea: a corba-based event architecture," in *COOTS'98: Proceedings of the 4th conference on USENIX Conference on Object-Oriented Technologies and Systems*. Berkeley, CA, USA: USENIX Association, 1998, pp. 9–9.

[12] J. van Hoof, "How eda extends soa and why it is important," 2.10.2008 2006.

[13] J. Pick, *Geo-Business: GIS in the Digital Organization*. Wiley, 2007.

[14] A. Kumar Harikumar, R. Lee, C.-C. Chiang, and H.-S. Yang, "An event driven architecture for application integration using web services," *Information Reuse and Integration, Conf, 2005. IRI -2005 IEEE International Conference on.*, pp. 542–547, Aug. 2005.

[15] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. Ferguson, *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR Upper Saddle River, NJ, USA, 2005.

[16] B. N. Bershad, C. Chambers, S. Eggers, C. Maeda, D. McNamee, P. Pardyak, S. Savage, and E. G. Sirer, "Spin - an extensible microkernel for application-specific operating system services," *SIGOPS Oper. Syst. Rev.*, vol. 29, no. 1, pp. 74–77, 1995.

[17] R. Rashid, D. Julin, D. Orr, R. Sanzi, R. Baron, A. Forin, D. Golub, and M. Jones, "Mach: a system software kernel," *COMPCON Spring '89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers.*, pp. 176–178, 27 Feb-3 Mar 1989.

[18] D. Bovet and M. Cesati, *Understanding the Linux Kernel*. O'Reilly Media, Inc., 2005.

[19] D. Solomon and M. Russinovich, *Inside Microsoft Windows 2000*. Microsoft Press Redmond, WA, USA, 2000.

[20] A. Chou, J. Yang, B. Chelf, S. Hallem, and D. Engler, "An empirical study of operating systems errors," *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 73–88, 2001.

[21] M. M. Swift, B. N. Bershad, and H. M. Levy, "Improving the reliability of commodity operating systems," *ACM Trans. Comput. Syst.*, vol. 23, no. 1, pp. 77–110, 2005.

[22] G. Miller, "The web services debate: .net vs. j2ee," *Commun. ACM*, vol. 46, no. 6, pp. 64–67, 2003.

[23] "Java 2 Platform, Enterprise Edition (J2EE)," SUN Microsystems, 4150 Network Circle, Santa Clara, CA 95054, Specification.

[24] *The .NET Framework*. Microsoft Corporation, 2004. [Online]. Available: \url{http://www.microsoft.com/net/}

[25] V. Stantchev, *Architectural Translucency*. Berlin, Germany: GITO Verlag, 2008.

[26] S. Frolund and J. Koistinen, "Quality of services specification in distributed object systems design," in *COOTS'98: Proceedings of the 4th conference on USENIX Conference on Object-Oriented Technologies and Systems (COOTS)*. Berkeley, CA, USA: USENIX Association, 1998. [Online]. Available: http://www.usenix.org/publications/library/proceedings/coots98/full_papers/frolund/frolund.pdf

[27] H. Ludwig, A. Keller, A. Dan, R. King, and R. Franck, "Web Service Level Agreement (WSLA) Language Specification," *IBM Corporation*, 2002.

[28] V. Tosic, K. Patel, and B. Pagurek, "WSOL-Web Service Offerings Language," *Web Services, E-Business, and the Semantic Web: CAiSE 2002 International Workshop, WES 2002, Toronto, Canada, May 27-28, 2002: Revised Papers*, 2002.

[29] D. Lamanna, J. Skene, and W. Emmerich, "SLAng: A Language for Defining Service Level Agreements," *Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems-FTDCS*, pp. 100–106, 2003.

[30] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web Services Agreement Specification (WS-Agreement)," *Global Grid Forum GRAAP-WG, Draft, August*, 2004.

[31] A. Polze and L. Sha, "Composite Objects: Real-Time Programming with CORBA," in *Proceedings of 24th Euromicro Conference, Network Computing Workshop, Vol.II, pp.: 997-1004*. Vaesteras, Sweden: Humboldt University of Berlin, Aug. 1998.

[32] W. Feng, "Dynamic client-side scheduling in a real-time corba system." in *COMPSAC*. IEEE Computer Society, 1999, pp. 332–333.

[33] P. Felber, R. Guerraoui, and A. Schiper, "Replication of corba objects." in *Advances in Distributed Systems*, ser. Lecture Notes in Computer Science, S. Krakowiak and S. K. Shrivastava, Eds., vol. 1752. Springer, 1999, pp. 254–276.

[34] V. Marangozova and D. Hagimont, "An infrastructure for corba component replication." in *Component Deployment*, ser. Lecture Notes in Computer Science, J. M. Bishop, Ed., vol. 2370. Springer, 2002, pp. 222–232.

[35] M. Werner, "Replikation in CORE, Bericht an das Graduiertenkolleg "Kommunikationsbasierte Systeme"," Oct 1996.

[36] P. Felber and P. Narasimhan, "Reconciling replication and transactions for the end-to-end reliability of corba applications." in *CoopIS/DOA/ODBASE*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 2519. Springer, 2002, pp. 737–754.

[37] P.-C. David and T. Ledoux, "An infrastructure for adaptable middleware." in *CoopIS/DOA/ODBASE*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 2519. Springer, 2002, pp. 773–790.

[38] S. Gutierrez-Nolasco and N. Venkatasubramanian, "A reflective middleware framework for communication in dynamic environments." in *CoopIS/DOA/ODBASE*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 2519. Springer, 2002, pp. 791–808.

[39] G. Biegel, V. Cahill, and M. Haahr, "A dynamic proxy based architecture to support distributed java objects in a mobile environment." in *CoopIS/DOA/ODBASE*, ser. Lecture Notes in Computer Science, R. Meersman and Z. Tari, Eds., vol. 2519. Springer, 2002, pp. 809–826.

[40] S. Adwankar, "Mobile corba," in *DOA '01: Proceedings of the Third International Symposium on Distributed Objects and Applications*. Los Alamitos, CA, USA: IEEE Computer Society, 2001, p. 52.

[41] V. Stantchev and M. Malek, "Architectural Translucency in Service-oriented Architectures," *IEE Proceedings - Software*, vol. 153, no. 1, pp. 31–37, February 2006.

[42] V. Stantchev and C. Schröpfer, "Service level enforcement in web-services based systems," *International Journal on Web and Grid Services*, vol. 5, no. 2, pp. 1741–1106, 2009.

[43] V. Stantchev and M. Malek, "Addressing Web Service Performance by Replication at the Operating System Level," in *ICIW '08: Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services*. Los Alamitos, CA, USA: IEEE Computer Society, June 2008, pp. 696–701.

[44] V. Stantchev, "Effects of Replication on Web Service Performance in WebSphere," International Computer Science Institute, Berkeley, California 94704, USA, ICSI Tech Report 2008-03, February 2008.

[45] V. Stantchev, T. D. Hoang, T. Schulz, and I. Ratchinski, "Optimizing clinical processes with position-sensing," *IT Professional*, vol. 10, no. 2, pp. 31–37, 2008.

[46] V. Stantchev and M. Malek, "Translucent replication for service level assurance," in *High Assurance Services Computing*. Berlin, New York: Springer, 06 2009, pp. 1–18.

[47] V. Stantchev, "Enhancing health care services with mixed reality systems," in *The Engineering of Mixed Reality Systems*. Berlin, New York: Springer, 09 2009.

[48] R. Berry, P. McKenney, and F. Parr, "Responsive systems: An introduction," *IBM Systems Journal*, vol. 47, no. 2, pp. 197–205, 2008.

[49] D. Bauer, L. Garce´s-Erice, S. Rooney, and P. Scotton, "Toward scalable real-time messaging," *IBM Systems Journal*, vol. 47, no. 2, pp. 237–251, 2008.

[50] N. Milanovic and M. Malek, "Current solutions for web service composition," *IEEE Internet Computing*, vol. 8, no. 6, pp. 51–59, 2004.

[51] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, *et al.*, "Business Process Execution Language for Web Services (BPEL4WS) 1.1," *Online: http://www-106. ibm. com/developerworks/webservices/library/ws-bpel, May*, vol. 139, p. 140, 2003.

[52] S. R. Ponnekanti and A. Fox, "Sword: A developer toolkit for web service composition," 2002. [Online]. Available: http://www.citebase.org/abstract?id=oai:wwwconf.ecs.soton.ac.uk:226

[53] F. Casati, S. Ilnicki, L.-J. Jin, V. Krishnamoorthy, and M.-C. Shan, "eflow: a platform for developing and managing composite e-services," *Research Challenges, 2000. Proceedings. Academia/Industry Working Conference on*, pp. 341–348, 2000.

[54] R. Meersman and Z. Tari, Eds., *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002 Irvine, California, USA, October 30 - November 1, 2002, Proceedings*, ser. Lecture Notes in Computer Science, vol. 2519. Springer, 2002.

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
issn: 1942-2679

International Journal On Advances in Internet Technology
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

International Journal On Advances in Life Sciences
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

International Journal On Advances in Networks and Services
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
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
issn: 1942-2628

International Journal On Advances in Systems and Measurements
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

International Journal On Advances in Telecommunications
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601