

Information Systems & Grid Technologies

Fourth International Conference ISGT'2010

Sofia, Bulgaria, May 28 – 29., 2010.

Proceedings



ISGT'2010 Conference Committees

Co-chairs

- Prof Ivan SOSKOV
- Acad Kiril BOYANOV

Program Committee

- Pavel AZALOV, Pennsylvania State University
- Marat BIKTIMIROV, Joint Supercomputer Center, Russian Academy of Sciences
- Marko BONAČ, Academic and Research Network of Slovenia
- Marco DE MARCO, Catholic University of Milan
- Milena DOBREVA, University of Strathclyde, Glasgow
- Viacheslav ILIN, Moscow State University
- Vladimir GETOV, University of Westminster
- Jan GRUNTORÁD, Czech Research and Academic Network
- Pavol HORVATH, Slovak University of Technology
- Arto KARILA, Helsinki University of Technology
- Dieter KRANZMUELLER, University of Vienna
- Vasilis MAGLARIS, National Technical University of Athens
- Ivan PLANDER, Slovak Academy of Science
- Dov TE'ENI, Tel-Aviv University
- Stanislaw WRYCZA, University of Gdansk
- Fani ZLATAROVA, Elizabethtown College

Organizing Committee

- Vladimir DIMITROV
- Maria NISHEVA
- Kalinka KALOYANOVA
- Vasil GEORGIEV

Vladimir Dimitrov Vasil Georgiev (Eds.)

Information Systems & Grid Technologies

Fourth International Conference ISGT'2010

Sofia, Bulgaria, May 28 – 29., 2010.

Proceedings

St. Kliment Ohridski University Press

Preface

This conference was being held for the fourth time in the end of May, 2010 in Sofia, Bulgaria. It is supported by the Scientific Research Fund of the University of Sofia "St. Kliment Ohridski" and by the Bulgarian Chapter of the Association for Information Systems (BulAIS). The Organizing committee consists of scientist from the Faculty of Mathematics and Informatics of the University of Sofia and from the Institute for Parallel Processing of the Bulgarian Academy of Sciences.

Total number of papers submitted for participation in ISGT'2010 was 39. They undergo the due selection by at least two of the members of the Program Committee. This book comprises 28 papers of 20 Bulgarian authors and 17 foreign authors included in one of the three conference tracks. A special section contains the selected publications of 5 young scientists. Responsibility for the accuracy of all statements in each peer-reviewed paper rests solely with the author(s). Permission is granted to photocopy or refer to any part of this book for personal or academic use providing credit is given to the conference and to the authors.

The editors

© 2010 Vladimir Dimitrov, Vasil Georgiev (Eds.)
ISBN 978-954-07-3168-1
St. Kliment Ohridski University Press

Table of Contents

Information Systems

Language Compatibility Issues in Multi-Party Web Services	9
<i>Sherimon Cherian, Reshmy Krishnan, Zlatinka Covacheva</i>	
Z-Specification of Relational Model of Data	18
<i>Vladimir Dimitrov</i>	
Implications of Preventing the Same Origin Policy (SOP) Loopholes on Web Mashups	30
<i>Vinu P. V., N. Girija</i>	
A Rule-Based Framework for Educational Board Games	39
<i>Boyan Bontchev</i>	
Regular Sparsity Map	51
<i>Ina Naydenova</i>	
SAS Data Warehouse and its Usage in Government Public Sector	64
<i>Snezana Savoska, Violeta Manevska, Silvana Kolevska</i>	
Business rules extraction and management - problems and solutions	72
<i>Evgeny Krustev</i>	
Modeling of Primary Copy Two-Version Two Phase Locking	79
<i>Svetlana Vasileva</i>	
Applying a Seven Streams Approach for Building a DW 2.0	83
<i>Nikola Hristov, Kalinka Kaloyanova</i>	

Distributed Systems and Grid Technologies

Ontology Design and Development for Grid Services	105
<i>Xhemal Zenuni, Florije Ismaili, Bujar Raufi</i>	
Early Evaluation of NPB UA Benchmark Scaling to Thousands of Blue Gene/P Cores Using PGAS-like OpenMP Extension	115
<i>Anton Korzh, Oxana Dzhosan</i>	
From Process Flow Diagrams to BPMN	125
<i>Vladimir Dimitrov</i>	
Load Balanced Resource Management for Cloud Systems	133
<i>Radko Zhelev, Vasil Georgiev</i>	
Sophie 2.0 – A Platform for Reading, Writing and Publishing of Electronic Books in a Network Environment	145
<i>Miloslav Sredkov, Trifon Trifonov, Kalin Georgiev</i>	

Grid framework for e-learning services	153
<i>Magdalena Todorova</i>	
Toward Semantic Specification of WS-BPEL	164
<i>Vladimir Dimitrov</i>	
Open MP parallel version of Loop's mesh subdivision algorithm	184
<i>Peter Armyanov</i>	

Intelligent Systems

Comparison of ACO behavior with various start strategies applied on MKP	191
<i>Stefka Fidanova, Pencho Marinov, Krassimir Atanassov</i>	
Providing and Maintaining Interoperability in Digital Library Systems	200
<i>Maria M. Nisheva-Pavlova</i>	
Catalogue Metadata in an Academic Digital Library	209
<i>Pavel I. Pavlov</i>	
Information Technologies for Presentation of Bulgarian Folk Songs with Music, Notes and Text in a Digital Library	218
<i>Lozanka Peycheva, Nikolay Kirov, Maria Nisheva-Pavlova</i>	
Modeling the process of decision making using Petri nets and multi-agent systems	225
<i>Elena Vlahu-Gjorgievska, Oliver Iliev, Nikola Rendevski, Natasa Blazevska- Tabakovska</i>	
Application of the Intranet in the Organizations at Knowledge Management	234
<i>Natasha Blazeska-Tabakovska, Violeta Manevska, Elena Vlahu-Gorgievska</i>	

Young Scientists Section

Development of Accounting Services for Grid	245
<i>Radoslava Goranova</i>	
Controlling presentation slides with tangible passive markers	254
<i>Svetoslav Neykov</i>	
Hello World in Parallel! An introductory lab exercises in parallel/concurrent and distributed programming	258
<i>Hristo Hristov</i>	
E-commerce Payment Systems	263
<i>Ana Serafimova</i>	
Popular Genealogy Software Products	273
<i>Darina Serafimova</i>	

Information Systems

Language Compatibility Issues in Multi-Party Web Services

Sherimon Cherian ^[1], Dr.Reshmy Krishnan ^[2], Dr.Zlatinka Covacheva ^[3]

Higher College of Technology, Muscat, Sultanate of Oman

^[1]pcsheri@yahoo.com, ^[2]reshmy_krishnan@yahoo.co.in, ^[3]zkovacheva@hotmail.com

Abstract. Web services are web-based applications developed using different programming tools that dynamically interact with other web applications using open standards that include XML, UDDI and SOAP. The web applications that involve calling multiple intermediary web services as part of an overall transaction are referred to as multi-party web services. There will be accessibility issues when accessing one web service with another one which is developed in another language because of the interoperability of the language. Because of the variety of web programming tools it cannot be avoided in accessing one web service developed in one language and another one developed in another language. For example there are issues in accessing a web service developed in Java language with another web service developed in PHP or C#. In this paper these issues have been discussed and some solutions have been suggested.

Keywords: Web Services, Java, C#, PHP, Multi-Party Web Services, Interoperability.

1 INTRODUCTION

Web services are self-contained, self-describing modular applications that can be published, located, and invoked across the Web. They represent an important evolutionary step in building Internet-based distributed applications. Web services allow buyers and sellers all over the world to discover each other, connect dynamically, and execute transactions in real time with minimal human interaction. At a minimum, web services can be any piece of software that makes itself available over the Internet using standardized web services messaging system and interface. The Web service technology comprises three basic standards:

- Web Services Description Language (WSDL): a format for describing the functionality of a service [5].
- Universal Description Discovery & Integration (UDDI): a registry that supports service publication and discovery [9].
- SOAP (formerly Simple Object Access Protocol): a protocol for exchanging messages among services.

XML, eXtensible Markup Language [4], lies at the core of web services interoperability, and enables it to provide a language-neutral and platform-independent way of linking applications [2]. Web services provide interoperability across platforms and languages. Organizations are developing web services in their



interests. It can be in J2EE or PHP or in other tools. But the client who is accessing the web service may be .NET client or Pearl client. For instance, making a call to a service checking the bank balance may in turn involve calls to multiple services before returning a response to the end user. Here each intermediary web services may be developed in different tools.

1.1 Web-services protocols

Web services are self-contained, modular applications that can be described, published, located, and invoked over the Internet. Web services architecture then requires three fundamental operations: publish, find, and bind. These operations along with the actors involved are shown in Figure 1 [3].

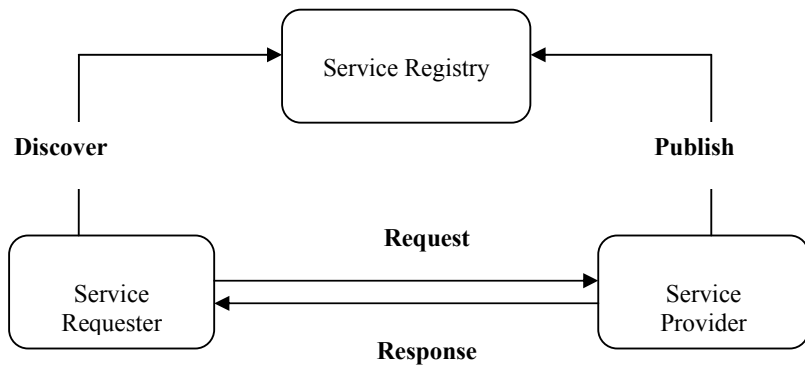


Figure 1: Web Services Actors and Operations

As shown in the figure, a service provider creates a web service and its service definition and then publishes it with a service registry based on UDDI specification. Once a service is published, a service requester may find the service via the UDDI interface. The UDDI registry provides the service requester with a service description based on WSDL. The service requester may then use this information to directly bind to the service and invoke it. All communication in the above scenario is based upon the standard messaging protocol, SOAP.

1.2 Service Oriented Architecture

Service Oriented Architecture (SOA) is a new architecture for the development of loosely coupled distributed applications. It is a collection of many services in the network. These services communicate with each other and the communication

involves exchange of data and even service coordination. Earlier SOA was based on DCOM or Object Request Brokers (ORBs). But nowadays SOA is based on web services [6].

SOA is much different from point-to-point architectures. SOA comprise loosely coupled, highly interoperable application services. These services can be developed in different development technologies (such as Java, .NET, C++, PERL, PHP etc), the software components become very reusable i.e. the same C# (C Sharp) service may be used by a Java application and / or any other programming language. WSDL defines a standard, which encapsulates the language specific implementation from the calling service.

1.3 Multiparty Web Service

The web applications that involve calling multiple intermediary web services as part of an overall transaction are referred to as **multi-party web services**. For instance, making a call to a service providing insurance quotes may in turn involve calls to multiple insurance provider services before returning a response to the end user.

1.4 Interoperable Web Services

An interoperable web service is one which can work across platforms, languages, applications and with web services from different vendors. Nowadays interoperable web services are inevitable part of web programming. As web services are shared software pieces the sharing of web services in different tools and platform are very important.

1.5 The need of Language Compatibility in Multiparty Web Services

Web Services are common and the user has the freedom to choose any web programming tool, platform, vendors to develop the web services. The interoperability can be done in several ways [7].

- Language Based: Web services developed in any tool like .NET, J2EE, PHP etc. should be able to integrate each other.
- Operating System Based: Web services developed in different operating systems like Solaris, Windows and Linux can co-exist and interoperate across assorted web environment.

Seamless integration will also enable greater collaboration for enterprises, by allowing them to leverage a larger ecosystem of partners in application development [8].

The paper is structured as follows: In Section 2, benefits & challenges to achieve interoperability are explained followed by suggestions to reduce language compatibility issues in the next section. Finally we discuss related work and conclude in Section 4 and Section 5, respectively.

2. PROFILE

In Service Oriented Architecture (SOA), and Web Services, Language Compatibility is an important factor in its success. It is generally accomplished by building the Web Services using the well-established guidelines for implementing Web Services and by following industry standards such as XML, WSDL, SOAP, and UDDI. But this is not sufficient to achieve the interoperability. The different products used for development also have to comply with many requirements such as the need to have similar implementations (data types, formats, and schemes) of the standards that you want to use. As different products are provided by different vendors, developed by several sets of people, and employ various types of underlying technologies, achieving a common understanding often becomes very difficult, which makes the products likely to be non-interoperable with each other. Over the last few years, the basic Web Services standards like XML, WSDL, and SOAP have matured a lot and WS-I have released a Basic Profile [12] that contains implementation guidelines for basic Web Services standards. Today, most vendors provide products that comply with the Basic Profile and support the standards included in the profile. With the wide adoption of the Basic Profile, software vendors have been able to make their products interoperable to a great extent.

There are four deliverables produced by the WS-I for the Basic Profile version 1.0. They are:

- Requirements and guidelines for writing interoperable Web services.
- Usage scenarios which describe fundamental ways in which providers and consumers interact.
- A sample application, which is an implementation of an interoperable Web service that demonstrates the requirements and guidelines presented in the Basic Profile.
- Testing tools which help developers to verify that their Web service implementations conform to the requirements in the Basic Profile.

Achieving interoperability for scenarios involving only basic standards is relatively easy if you follow the guidelines set by the Basic Profile (BP) 1.0 or 1.1 of the Web Services Interoperability Organization (WS-I) [12]. The Basic Profile consists of

implementing guidelines recommending how a set of core Web Services specifications should be used together to develop interoperable Web Services. The guidelines address technologies that cover four core areas: Messaging, Description, Discovery, and Security.

2.2 Benefits of Interoperability

The benefits can be summarized as follows:

- Facilitate the creation of a new presentation layer using a new technology and, at the same time, reuse the existing business components
- Integrate heterogeneous software components within an enterprise
- Lower Integration efforts
- Lower Ownership and Maintenance efforts. The cost and efforts of ownership and maintenance inherently decrease with the use of interoperable components, supplied by third parties
- Increased market and technology opportunities. With the scope of web services and interoperability, the enterprises have a wider choice of vendors, as technologies no longer remain a hindering factor

2.3 Challenges to Interoperability

As is the case with any technologies, the web service and the interoperability aspects also present the providers and consumers with a few bottlenecks. The primary reason for most of these challenges is that web service is an emerging technology. Some of the challenges faced include the following:

- One of the most intriguing challenges in interoperability is the mismatch between the data types of "interoperating" platforms. Other languages may not support the data types supported by one language. Though there may be equivalent data types, it is difficult to find interoperable data types. For example, the Arraylist type in Java is equivalent to the ArrayList type in Microsoft.NET though one cannot say they are directly interoperable. This is due to the fact the implementation of these types are specific to their parent platforms [11]
- Every organization tries to address the interoperability in their own perspective, giving rise to "proprietary" standards, which defeat the very objective of web services and interoperability. These specifications cannot be guaranteed to be compatible with those adopted by other vendor(s) or client(s)
- Lack of any industry standards and/or specifications

- With the growing scope and horizon of the web services, new specifications and standards are being constantly added to the existing set by the vendors, consortia of companies, standard bodies as well as individual companies. These groups may or may not be communicating with each other regarding any of their releases. This contributes to a growing set of incompatible specifications

The best way to counter the diverse specifications and surmounting barriers to the web services interoperability is through consensus and industry-accepted and guided specifications. This shall be the only way that the enterprise end-users are assured of interoperable web service solutions.

3. SUGGESTIONS FOR DEVELOPING WEB SERVICES TO REDUCE LANGUAGE COMPATIBILITY ISSUES

The following are some suggestions to solve the language compatibility issues between web services developed in Java and in Microsoft .NET.

Check for Empty Arrays

If an array of objects is sent over a web service, always ensure that the array contains valid data [10]. Sending empty arrays over web services can create issues because some tools consider an empty array as a single null value.

Generation of Client Proxies

Many Java based tools have the option of specifying a unique package and type name when generating client proxies. This is essential when you are creating proxies for Web Services that share the same data type.

Java Beans

Ensure that an object is not null when an IDE is used to generate Java Beans from an XSD file.

Null Dates and Times

In Java, `java.util.Date` and `java.util.Calendar` are classified as reference types. In .NET, `System.DateTime` is considered a value type. Reference types can be null, whereas value types cannot. If you are planning to send null date values across a Web Service, always send the value in a complex type, and set the value of the complex type to null. This will help prevent the null date value being interpreted incorrectly (and raising an exception).

compareTo() method

If sending dates and times over a Web Service between .NET and Java, always use the appropriate compareTo() method in Java to compare dates (as opposed to date == value). This will help ensure accuracy for date comparisons between the platforms.

Trace Tool

A Trace Tool can be invaluable for investigating SOAP requests and responses between Web Services. It can help validate data types and the construct of the message, and also report SOAP faults that you may miss in a browser. Using one that intercepts the request is more difficult to setup, but easier to use than looking through trace files.

Option to Change Host and Port

When designing your Web Service client, consider adding a helper method to change the host and port values of the Web Service location. This makes it easy if the location of the Web Service changes in the future or you want to redirect the output to a trace tool.

Ensure Document/Literal when generating Web Services

Many toolkits have the option of choosing either RPC/Encoding or Document/Literal as the default format when generating Web Services. To help ensure compliance with the WS-I Basic Profile 1.0, select Document/Literal as the default encoding mechanism for all of your Web Services.

Use Unit Tests to Test Interoperability

Unit tests (using NUnit for .NET and JUnit for Java) are invaluable for testing interoperability of multiple data types over a Web Service. Re-running the tests if data types change (or if you change versions of Web Services toolkits!) will give you the confidence that the Web Services that you design are fully interoperable.

Use XSD First

When designing for interoperability, always start with defining the data first. Deciding on what data will be sent, creating the data type in XSD first, and then using tools to generate classes from the XSD file will help guarantee data type interoperability.

4 RELATED WORK

It was the challenges, as described above, that forced the technology leaders in Web Service specifications to collaborate and create the Web Services Interoperability Organization (WS-I) [12]. The organization was founded in 2002, and is aimed at promoting interoperability across platforms, applications and languages. The charter of the WS-I is to accelerate the adoption of Web Services by assisting in the selection and interpretation of WS specifications and in the development of common best practices for development and deployment and integration of business applications.

The organizations joining the WS-I can benefit from the following perspectives:

- Since the WS-I is collaborated on the basis of use-cases derived from industry specifications, it acts as a forum for communicating the business requirements and those for distributed computing interoperability.
- The organizations having proven credentials in a particular vertical, like Insurance and Hospitality, can help leverage and augment the horizontally focused computing with vertically focused business requirements.

Apart from the access to WS-I profiles, the member also gains access to the testing methodologies for web services, the usage & implementation scenarios and use-cases for the same. All these contribute to a reduced cost in development, deployment and integrations of their service-oriented solutions.

5. CONCLUSION

Web services greatly help developers build highly integrated solutions. Web services are certainly going to form a huge chunk of the enterprise deployments in near future. The deciding factor in their success and acceptability will be the extent of interoperability provided to the enterprise partners. To achieve this, a cohesive interpretation of ambiguities, incompatibilities and best practices is needed. These are precisely the lines that the WS-I is functioning on. In this paper issues involved in web services interoperability between Java and Microsoft .NET are discussed and some solutions have been suggested. So in order to keep Web services interoperable, we need to ensure that various vendor implementations are interoperable, the tools that are used do not affect the integrity of the Web services, and standards are strictly followed and implemented.

REFERENCES

- [1] Chris Kaler (Editor). WS-Security, Version 1.0. An IBM, Microsoft and VeriSign joint specification. April 5, 2002.
<http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>

- [2] Don Box (DevelopMentor) SOAP: Simple Object Access Protocol 1.1 W3C Note. May 8, 2000.
<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>
- [3] K.Gottschalk, S. Graham, H. Kreger, and J. Snell Introduction to Web services architecture. IBM Systems Journal, Volume 41, Number 2, 2002
- [4] Jayavel Shanmugasundaram¹, Eugene Shekita¹, Rimón Barr², Michael Carey³, Bruce Lindsay¹, Hamid Pirahesh¹, Berthold Reinwald¹ Efficiently publishing relational data as XML documents , The VLDB Journal (2001) 10: pp.133–154
- [5] Eyhab Al-Masri and Qusay H. Mahmoud Department of Computing and Information Science University of Guelph, Guelph, ON, N1G 2W1 Canada, Investigating Web Services on the World Wide Web, WWW 2008 Web Engineering - Web Service Deployment Beijing, China
- [6] Service Oriented Architecture, <http://www.service-architecture.com/>
- [7] <http://www.sun.com/smi/Press/sunflash/2005-11/sunflash.20051104.1.xml>
- [8] Common Language Resources and Technology Infrastructure, <http://www.clarin.eu>
- [9] UDDI Version 3.0.2. http://www.uddi.org/pubs/uddi_v3.htm.
- [10] Arun Gupta, Project Tango: Adding quality of service and .NET Interoperability to the Metro Web Services stack: <http://blogs.sun.com/arungupta>
- [11] Narayana Rao Surapaneni, Dhananjay Khatre, “Java and .NET a developers guide to Interoperability and Migration, Prentice Hall India”
- [12] Web Services Interoperability Organization. <http://www.ws-i.org/>

Z-Specification of Relational Model of Data

Vladimir Dimitrov¹,

¹ University of Sofia, Faculty of Mathematics and Informatics, James Bourchier Blvd. 5,
1164 Sofia, Bulgaria
cht@fmi.uni-sofia.bg

Abstract. This paper is an attempt to specify formally the Relational Model of Data with NULL values and named columns. The specification could be further extended to capture Object-Relational Model of Data and Data Streams.

Keywords: Relational Model of Data, Z-Notation, Formal Specification.

1 Introduction

The Relational Model of Data (RMD) is specified here following its original presentation given in [1]. Z-notation is used as formal notation. It is now an international standard [2]. The original presentation of RDM is not formalized, nor detailed or consistent. It contains many open topics, which have been solved later sometimes in different ways in RMD implementations (for dummies: see, for example, how RMD is implemented in DB2 and Oracle). In a formal specification such topics could not be evaded. The investigation on the topic shows that there is only one attempt in that direction [3]. It is a Master Thesis that is very huge and impractical to be used as a basis for any extensions.

2 Schemas

The main types in this specification are:

$[R\text{NAMES}, C\text{NAMES}, \text{COMPONENTS}]$

$VALUES ::= NULL \mid noNull \square \text{COMPONENTS} \square$

where RNames is the set of all possible relation names, CNames is the set of all possible column names, COMPONENTS is the set of all non NULL component values, and VALUES is the set of all possible tuple component values. So, the set of domains is specified as a set of finite non-empty sets of values:

$DOMAINS == \square_1 VALUES$

Relation columns are named and have domains as specified:



COLUMNS

name: C NAMES

domain: DOMAINS

The relation schema is a non-empty finite set of columns:

$$SCHEMAS == \text{seq}_1 COLUMNS$$

Domains are non-empty, finite sets of values and schemas are non-empty, finite sequences of columns (domains). In such a way, the problems that arise from endless domains and schemas are overcome.

Database schema is defined as a partial function from relation names into schemas:

DBSchema

db: R NAMES \square *SCHEMAS*

Database schema describes the state of all relational schemas. The initial state of database schema is a function with an empty domain, in other words, there are no relation names bounded with schemas:

DBSchemaInit

DBSchema

$\text{dom } db = \emptyset$

Relation schema size is returned by:

DBSRelationSchemaSize

$\exists DBSchema$

n?: R NAMES; size!: \mathbb{N}_1

$n? \in \text{dom } db \wedge \text{size!} = \#(db(n?))$

The database schema state is changed when new relation schemas are added; old ones are removed, or relation is renamed:

DBSRelationSchemaSize

$\exists DBSchema$

n?: R NAMES; size!: \mathbb{N}_1

$n? \in \text{dom } db \wedge \text{size!} = \#(db(n?))$

The database schema state is changed when new relation schemas are added; old ones are removed, or relation is renamed:

DBSAdd

$\Delta DBSchema$

$n?: RNames; s?: SCHEMAS$

$n? \notin \text{dom } db \wedge db' = db \cup \{n? \mapsto s?\}$

DBSRemove

$\Delta DBSchema$

$n?: RNames$

$n? \in \text{dom } db \wedge db' = \{n?\} \sqcap db$

DBSRename

$\Delta DBSchema$

$n?: RNames; new?: RNames$

$n? \in \text{dom } db \wedge new? \notin \text{dom } db \wedge$
 $db' = \{n?\} \sqcap db \cup \{new? \mapsto db(n?)\}$

The database schema state changes, when in a relation schema: a column name is renamed, a column domain is changed, a column is removed, or a new column is added:

DBSRenameColumn

$\Delta DBSchema$

$n?: RNames; i?: \mathbb{N}_1; new?: CNames$

$n? \in \text{dom } db \wedge i? \in 1.. \#(db(n?)) \wedge$

$(\exists c: COLUMNS \sqcap c.name = new? \wedge$

$c.domain = ((db(n?))(i?)).domain \wedge$

$db' = db \oplus \{n? \mapsto (db(n?) \oplus \{i? \mapsto c\})\}$

DBSChangeDomain

$\Delta DBSchema$

$n?: RNames; i?: \mathbb{N}_1; new?: DOMAINS$

$n? \in \text{dom } db \wedge i? \in 1..\#(db(n?)) \wedge$

$(\exists c: COLUMNS \square c.name = ((db(n?))(i?)).name \wedge$

$c.domain = new? \wedge$

$db' = db \oplus \{n? \mapsto (db(n?) \oplus \{i? \mapsto c\})\}$

DBSRemoveColumn

$\Delta DBSchema$

$n?: RNames; i?: \mathbb{N}_1$

$n? \in \text{dom } db \wedge i? \in 1..\#(db(n?)) \wedge$

$db' = db \oplus \{n? \mapsto ((1..(i?-1)) \cup ((i?+1).. \#(db(n?)))) \uparrow db(n?)\}$

DBSInsertColumn

$\Delta DBSchema$

$n?: RNames; i?: \mathbb{N}_1; c?: COLUMNS$

$n? \in \text{dom } db \wedge i? \in 1..\#(db(n?)) \wedge$

$db' = db \oplus$

$\{n? \mapsto ((1..(i?-1)) \uparrow db(n?)) \hat{\ } \square c? \square \hat{\ } ((i?.. \#(db(n?))) \uparrow db(n?))\}$

These operations impact the current relation instance. Relation schema update is a very heavy operation and could be conflicting with some other RMD operations, which are still not introduced here.

3 Relation Instance

The instance of a relation is a set of tuples – possibly empty set. The set of all possible tuples has to be specified:

$TUPLES == seq_1 VALUES$

$RELATIONS == \square TUPLES$

These relation instances have no links to its schema and they are not restricted, so to do that database instance is defined as:

DBInstance

DBSchema

instance: RNames \square *RELATIONS*

$\text{dom } instance = \text{dom } db \wedge (\forall n: RNames \square n \in \text{dom } instance \wedge$

$(\forall t: TUPLES \square t \in instance(n) \Rightarrow \#t = \#(db(n)) \wedge$

$(\forall i: 1..\#t \square t(i) \in ((db(n))(i)).domain))$

Initially, database instance is empty:

DBInstanceInit

DBInstance

$\text{dom } instance = \emptyset$

Now, following SQL terminology, relation instances are created, dropped and renamed:

DBCreate

$\Delta DBInstance$

DBSAdd

$instance' = instance \cup \{n? \mapsto \emptyset\}$

DBDrop

$\Delta DBInstance$

DBSRemove

$instance' = \{n?\} \square instance$

DBRename

$\Delta DBInstance$

DBSRename

$$instance' = (\{n?\} \sqsubseteq instance) \cup \{new? \mapsto instance(n?)\}$$

Changes in relation schema impact on relation instance:

DBRenameColumn

$\Delta DBInstance$

DBSRenameColumn

$$instance' = instance$$

DBChangeDomain

$\Delta DBInstance$

DBSChangeDomain

$$\text{dom } instance' = \text{dom } instance \wedge (\forall n: RNames \sqsubseteq$$

$$n \in \text{dom } instance \wedge$$

$$(n \neq n? \wedge instance'(n) = instance(n) \vee$$

$$n = n? \wedge instance'(n?) = \{t: TUPLES \mid \forall u: TUPLES \sqsubseteq$$

$$u \in instance(n?) \wedge$$

$$t = ((1..(i?- 1)) \uparrow u) \hat{\sqsubseteq} NULL \hat{\sqsubseteq} ((i?.. \#(db(n?))) \uparrow u)\})$$

DBInsertColumn

$\Delta DBInstance$

DBSInsertColumn

$\text{dom } instance' = \text{dom } instance \wedge (\forall n: RNames \square$

$n \in \text{dom } instance \wedge$

$(n \neq n? \wedge instance'(n) = instance(n) \vee$

$n = n? \wedge instance'(n?) = \{t: Tuples \mid \forall u: Tuples \square$

$u \in instance(n?) \wedge$

$t = ((1..(i?- 1)) \uparrow u) \wedge \square NULL \square \wedge ((i?.. \#(db(n?))) \uparrow u)\})$

Tuples could be added or removed to/from relation instance, but their components have to be members of the corresponding domains of the relation. This condition has to be satisfied by every relation instance. When tuples are added or removed from a relation instance, they have to be checked in advance “Is the tuple possible in that relation?”, and this is done with the supporting Z-schema CHECK that is used later in the corresponding Z-schemas that add and remove tuples:

CHECK

$db: RNames \square SCHEMAS$

$n: RNames; t: Tuples$

$\#t = \#(db(n)) \wedge (\forall i: 1..\#t \square t(i) \in ((db(n))(i)).domain)$

DBInsert

$\Delta DBInstance$

$n?: RNames; t?: Tuples$

$CHECK[n?/ n, t? / t] \wedge t? \notin instance(n?) \wedge$

$\text{dom } instance' = \text{dom } instance \wedge$

$(\forall n: RNames \square n \in \text{dom } instance \wedge$

$n \neq n? \Rightarrow instance'(n) = instance(n) \wedge$

$instance'(n?) = instance(n?) \cup \{t?\}$

DBDelete

DBInstance

$n?: RNames; t?: Tuples$

$CHECK[n?/n, t?/t] \wedge t? \in instance(n?) \wedge$

$dom\ instance' = dom\ instance \wedge$

$(\forall n: RNames \square n \in dom\ instance \wedge$

$n \neq n? \Rightarrow instance'(n) = instance(n)) \wedge$

$instance'(n?) = instance(n?) \setminus \{t?\}$

Operations on relation instance could be done by keys. First, superkeys and keys are defined, and then corresponding operations. Note, that tuple update is now restricted not to be done on key columns:

SuperKeys

DBInstance

$n?: RNames; superKeys!: \square_1 (\square_1 \mathbb{N}_1)$

$n? \in dom\ db \wedge superKeys! = \{k: \square_1 \mathbb{N}_1 \mid (\forall i: \mathbb{N}_1 \square$

$i \in k \Rightarrow i \leq \#(db(n?))) \wedge$

$(\forall t1, t2: Tuples \square t1 \in instance(n?) \wedge t2 \in instance(n?) \wedge$

$k \triangleleft t1 = k \triangleleft t2 \Leftrightarrow t1 = t2)\}$

Keys

DBInstance

$n?: RNames; keys!: \square_1 (\square_1 \mathbb{N}_1)$

$\exists_1 superKeys: \square_1 (\square_1 \mathbb{N}_1) \square$

$SuperKeys[superKeys / superKeys!] \wedge keys! \subseteq superKeys \wedge$

$(\forall k: keys! \square$

$\neg(\exists subset: \square_1 \mathbb{N}_1 \square subset \subset k \wedge subset \in superKeys))$

CHECKKeyAndKeyValue

DBInstance

$n?: RNAMEs; key?: \square_1 \mathbb{N}_1; keyValue?: TUPLES$

$\exists_1 keys: \square_1 (\square_1 \mathbb{N}_1) \square Keys[keys / keys!] \wedge$

$key? \in keys \wedge \#key? = \#keyValue? \wedge$

$(\forall i: key? \square keyValue?(i) \in ((db(n?))(i)).domain)$

DBDeleteByKey

$\Delta DBInstance$

$n?: RNAMEs; key?: \square_1 \mathbb{N}_1; keyValue?: TUPLES$

$CHECKKeyAndKeyValue \wedge \text{dom } instance' = \text{dom } instance \wedge$

$(\forall n: RNAMEs \square n \in \text{dom } instance \wedge$

$n \neq n? \Rightarrow instance'(n) = instance(n)) \wedge$

$((\exists_1 t: TUPLES \square t \in instance(n?) \wedge$

$(\forall i: key? \square t(i) = keyValue?(i)) \wedge$

$instance'(n?) = instance(n?) \setminus \{t\}) \vee$

$\neg(\exists_1 t: TUPLES \square t \in instance(n?) \wedge$

$(\forall i: key? \square t(i) = keyValue?(i))) \wedge$

$instance'(n?) = instance(n?)$

CHECKKeyAndAttributes

DBInstance

$n?: R\text{NAMES}; key?: \square_1 \mathbb{N}_1; keyValue?: TUPLES;$

$atts?: \square_1 \mathbb{N}_1; attsValue?: TUPLES$

$$\begin{aligned} & \exists_1 keys: \square_1 (\square_1 \mathbb{N}_1) \square Keys[keys / keys!] \wedge \\ & key? \in keys \wedge \#key? = \#keyValue? \wedge \\ & (\forall i: key? \square keyValue?(i) \in ((db(n?))(i)).domain) \wedge \\ & atts? \subset 1..\#(db(n?)) \wedge (\forall k: keys \square k \cap atts? = \emptyset) \wedge \\ & \#atts? = \#attsValue? \wedge \\ & (\forall i: atts? \square attsValue?(i) \in ((db(n?))(i)).domain) \end{aligned}$$

DBUpdate

$\Delta DBInstance$

$n?: R\text{NAMES}; key?: \square_1 \mathbb{N}_1; keyValue?: TUPLES;$

$atts?: \square_1 \mathbb{N}_1; attsValue?: TUPLES$

$$\begin{aligned} & CHECKKeyAndAttributes \wedge \text{dom } instance' = \text{dom } instance \wedge \\ & (\forall n: R\text{NAMES} \square n \in \text{dom } instance \wedge \\ & \quad n \neq n? \Rightarrow instance'(n) = instance(n)) \wedge \\ & ((\exists_1 t: TUPLES \square t \in instance(n?) \wedge \\ & \quad (\forall i: key? \square keyValue?(i) = t(i)) \wedge \\ & \quad instance'(n?) = instance(n?) \setminus \{t\} \cup \\ & \quad \{u: TUPLES \mid \#u = \#t \wedge (\forall i: 1..\#t \square \\ & \quad \quad i \in atts? \wedge u(i) = attsValue?(i) \vee \\ & \quad \quad i \notin atts? \wedge u(i) = t(i))\}) \vee \\ & \neg(\exists_1 t: TUPLES \square t \in instance(n?) \wedge \\ & \quad (\forall i: key? \square keyValue?(i) = t(i)) \wedge \\ & \quad instance'(n?) = instance(n?)) \end{aligned}$$

4 Conclusions and Future Work

What is not put in this specification? When columns are removed no checks for keys are defined by the RMD and in the specification this is not done, but in an implementation this has to be resolved.

What more can be done with this specification? Originally, in [1] relational algebra is specified as operations on RMD, so this specification could be extended to support it.

In the next following, I will try to answer some questions that have been put on the topics of this paper from some referees at a local conference, which even has some ambitions to be “international”.

Cite: *“The section of related works could be extended. It is not clear why the author uses Z-notation.”* Intention of this paper is to specify RMD with Z-notation, but “Why in Z-notation?” - Because it is an internationally accepted mature standard suitable for that. Standard notations are clear for all; they have no deviations and specializations forcing the reader to learn the notation before starting to read the specification.

Cite: *“The motivation of formalization of a relation data model should be improved. What do you mean presentation? Nowadays, there are tools, which allow generating code of a database model, like PowerDesigner, Oracle and etc., from a relational data model. Therefore, I believe that presentations of a relational data model, used in these tools, are formal.”* First, the formal specification of RMD is a high level specification of the RMD, applicable, possibly, to all systems that implement it. Second, in [1], RMD is not specified formally, which means that it is not specified with a formal notation of some formal system. The power of formal systems is that it is possible to prove formally by reasoning some properties of the specified system, even if the last one is very complex. Third, above mentioned tools are using implementations of RMD: the development process consists: from specifications to implementations, and it is very important before starting to implement to verify the specification.

Cite: *“What is application of the proposed specification? Do we need it, if it is not applicable?”* This specification is not valuable for persons who do not differentiate between specification and implementation. But let try to answer the question as for dummies. This specification could be extended and elaborated for real implementations of RMD, like DB2 and Oracle, and then used in tools for program code verification, more precisely, for proving given properties (requirements) put on the persistent data.

Cite: *“Motivation is missing. Specifications may be useful as such, but they also support automatic program derivation. In this case the final goal could be not less than generation of a relational DBMS! However, this is probably unrealistic. Verifying the correctness and consistency of the specifications would also be very demanding.”* It is great! Specifications are for automatic program generation and in the case for RDBMS generation! Some people do not know what is DBMS, how complicated it is to think for it as a simple program. This paper does not solve all the problems for program verification, but it is a step in that direction.

Finally, in short publications could be published only a part of the whole work. The formal specification of RMD is a hard work that could not be done in a short time.

Some of my earlier publications have covered only a part of it, but introduction of new concept forced me to reevaluate this earlier specifications. No doubts, that it will happen in the future with the new extensions, but that is the game.

Acknowledgments. This research is supported by Project VU-MI-204/2006 “Intelligent Sensor Systems for Security Enhancement” funded by the Bulgarian Science Fund.

References

1. Codd, E. F., A Relational Model of Data for Large Shared Data Banks, CACM, vol. 13, No. 4, June, 1974, pp. 377-387.
2. ISO/IEC 13568 : 2002 (E) Information Technology. Z Formal Specification Notation. Syntax, Type System and Semantics, www.iso.org.
3. Baluta, D. D., A Formal Specification in Z of the Relational Data Model, Version 2 of E. F. Codd, Concordia University, Montreal, Quebec, Canada, March 1995.

Implications of Preventing the Same Origin Policy (SOP)

Loopholes on Web Mashups

Vinu P.V.¹, Dr. N.Girija²

Higher College of Technology, Muscat, Sultanate of Oman

¹vinusheri@yahoo.co.uk, ²nbgir2004@yahoo.co.in

Abstract. A web mashup is a web application that gathers data from several sources around the web integrates it and displays the result to user. Recently web mashups are becoming more popular; hence the problem of securing the information becomes very significant. Same-Origin Policy (SOP) is the current security model implemented by majority of the browsers. The basic idea of this policy is not to allow content from different sources to be used in a single document. Currently available web browsers are not designed to support web mashups. Therefore the current security model is insufficient for mashup applications. And SOP seems to be a source of many security problems that mashups face nowadays. There are two important security issues with mashups. One is security of the technologies used to create mashups and the other one is trustworthiness of the contents of mashups. This paper focused on how to prevent SOP loopholes on web mashups and also reviewed the currently available solutions to mashup security and proposed a suitable enhanced solution for this challenge.

Keywords: web mashups, same-origin policy, mashup security, cross-site scripting, cross-site request forgery.

1 INTRODUCTION

Web Applications collect data from various sources available from the Internet and integrates that data and creates a new application called web mashups [20]. For example, one web site is providing only map of a city and another web site is providing other information about the city such as location of hotels, hospitals, tourist spots etc. A Web mashup application combines both the websites; the resultant web mashup application provides map plus information of the city in a single application. The first known mashup began with the API provided by Google for Google Maps and then for Google Earth. The advent of Web 2.0 fostered a transition from desktop applications to web applications. The Yahoo, Flickr facilitates the user to develop their own web application which is useful for storing and manipulating data for their custom based own style. All these existing services can be integrated to build a single application [2]. Mashup introduce



new types of threats because of their dynamic and multi-domain nature. Current methods of making mashups, which combine data and tools from multiple sources on the Web, introduce security concerns. As mashups grow in popularity, the problem of securing information between mashup components becomes increasingly important. Web sites today either sacrifice security for functionality, or sacrifice functionality for security.

1.1 Web Mashups

The key driving power of the Internet is information, and mashup is an application that takes that information and presents it in a unique way. A Web Mashup is a web application that integrates content from different providers to create a new service, not offered by the content providers [4]. Google Maps is a very popular interface to use in mashups. Even though the idea of mashups is combining data from two or more sources, some mashups use data from a single source. A good example of this is TwitterSpy, which only pulls data from Twitter [9]. Web mashups and distributed portal, both gets information from other websites, but a web mashup differs from a distributed portal in that it creates something new from the information [7]. The number of web mashups is rapidly increasing. <http://www.programmableweb.com> [1] is a site that gives updated information of registered Web mashups currently available. This site contains more than 4000 registered mashups [4].

1.2 Same Origin Policy (SOP)

The same-origin policy was originally released with Netscape Navigator 2.0. It is a key security measure imposed by internet browser developers to protect the data and scripts of one page from being accessed by another page coming from a different origin [3]. Under the “Same Origin Policy”, a web resource can only interact with another web resource if and only if both resources are from the same origin. It prevents websites from one domain from requesting data belonging to another domain. Here by the clause “same origin” it means that the domain name, port, and protocol should be the same for both pages [11]. This policy is one of the foundations of Web security and is implemented by all popular Web browsers. Theoretically, it is very simple, but the restrictions it applies to web applications can be quite delicate. Documents from the same origin may freely access each other's content, but applications from different domains are isolated from each other

since different domains represent different originators. As a result, if applications in multiple windows or frames are downloaded from different servers, they will not be able to access each other's data and scripts [5]. The nature of mashups is to use content from different sources, but the nature of SOP is not to allow using content from different sources. So the conflict arises. Here the problem is that when the website developer embeds code written by a third party on her site, the same-origin policy no longer offers any protection, and the embedded code likely has access to information stored on the developer's site. It's not easy to mash content from multiple sources, when everything must be squeezed through the originating web server. Because of the nature of mashup, security problems exist. Mashup applications are built using data combined from one or more internal and external sites, typically in an ad hoc manner. Whenever a site integrates third-party content, the site runs the risk of incorporating malicious content [20]. This model of development leads to several security vulnerabilities. Traditional security measures such as DMZs [16] and firewalls are not enough to address mashup security problems. With each new data source added to a mashup, security vulnerabilities increase. So it is essential to ensure that data are secured against malicious attacks. Also users need to make sure the content they see in the mashup is reliable, especially when it is coming from an un-known blog or an article that anyone can edit [17].

The paper is structured as follows: In Section 2, exceptions of SOP is explained followed by the discussion of security vulnerabilities in web mashups in the next section. Section 4 describes the recommendations to improve mashup security. Finally we discuss related work and conclude in Section 5 and Section 6, respectively.

2. EXCEPTIONS OF SOP

The same origin policy applies to only HTML documents. But there are some exceptions. Resources like javascript files and images (, <script > etc) can be imported into an HTML page from other domains. After importing, such resources are considered to be of the same origin as the integrated document [4]. Also the domain path is only considered to check the origin, not sub-directories. For e.g., <http://www.xyz.com/~joe> is recognized as the same origin as <http://www.xyz.com/~jack> but may be these directories belong to different owners. Also for example <http://www.google.com> and 74.14.205.100 are considered as different domains even if the IP address of the above site is 74.14.205.100. Also the policy does not apply to hosted contents [3] [5]. Web mashups fetch information from various untrusted sources such as posts, blogs, search results etc. Most of the mashup contents are contributed mostly by anonymous users. This content is never validated prior to being served to the browser. So even if the Web server is

trustworthy, there is no guarantee in the contents hosted by these servers. It may include malicious code [5]. Also a script can set the value of `document.domain` to a suffix of the current domain [12]. In that case, the shorter domain is used for subsequent origin checks. For example, assume a script in the document at `http://A.ABC.com/abc.html` executes the statement: `document.domain = "ABC.com"`; after execution of that statement, the page would pass the origin check with `http://ABC.com/xyz.html`. So these exceptions are available for hackers to exploit. In fact, the only time same-origin explicitly restricts document retrieval is when the `XmlHttpRequest` method is used. `XmlHttpRequest` is the underlying technology that drives web mashups.

A mashup application suffers from cross-site request forgery (CSRF) attacks, Asynchronous JavaScript + XML (Ajax) vulnerabilities, cross-site scripting (XSS), and other potential security weaknesses [11]. Of these, CSRF and XSS are the two main security vulnerabilities.

3 SECURITY VULNERABILITIES IN WEB MASHUPS

3.1 Cross-site scripting (XSS)

Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web applications that enables malicious attackers to inject client-side script into web pages viewed by other users [10]. The person behind the attack succeeds in executing a script on the user's browser making it feel that the script is coming from a trusted web site [6]. It is a common attack in which an attacker injects a malicious piece of code into a web site. Some examples where XSS is used are to steal cookie values, to access restricted information, to rewrite parts of a web page etc. The two basic types of XSS attacks are:

- Reflected XSS
- Stored XSS

In reflected XSS, a user is somehow lured to send malicious code to server. The server in

turn responds and prints the user input as content of response page. The malicious code is “reflected” back to user as page content. Browser executes the code as a part of trusted content. This type of attack exploits vulnerable Web applications that display input parameters back to the browser without checking for the presence of active content in them.

In stored XSS attack, malicious codes is inserted and stored in forums, discussion groups or in post your comments section of a website. The result is that the malicious codes along with the user input will be executed by the browser. This type of attack has become more important with the popularity of Web 2.0. Web 2.0 promotes sharing, interaction, and collaboration among people, and so users have more chance of seeing other (potentially malicious) users' input through services such as discussion groups, or blogs.

The underlying problem is that many Web pages display input that is not validated. If input is not validated, malicious script can be embedded within the input. If a server-side script then displays this non-validated input, the script runs on the browser as though the trusted site generated it.

3.2 Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) is an exploit in which the hacker relies on vulnerabilities in both the browser and the target web site [3]. Most web applications require user authentication, which is carried out by using cookies or HTTP authentication. Most of the web sites trust the users so that the inputting of user id and password is not needed in every connection [site remembers the user]. CSRF exploits this trust.

Unlike XSS, CSRF does not require malicious script to be injected into a trusted Web mashup page. Imagine that a browser starts an authenticated session between a mashup page and a mashup server. After that the server assumes that all the requests coming from the browser are valid, since the session is authenticated. Now an intruder site can issue requests to a mashup server through the authenticated mashup page. Server responses are then transmitted unknowingly back to the intruder site. This is CSRF attack. Typical

authentication mechanisms such as user name/password, cookies, and SSL certificates are not sufficient for protecting against CSRF attacks, because these mechanisms depend on authenticating sessions between a browser and the server, not between each individual request between the browser and the server.

4 RECOMMENDATIONS

Web mashups are applications that integrate data from different sources. This integration can lead to several security vulnerabilities. So it is essential to ensure the correctness of data before including it in a mashup environment. The DMZ [16] and firewalls are traditional security measures used to secure data against malicious attacks. But they are inefficient to meet the security needs required for web mashups.

Input validation is the first step towards protecting Web applications. An effective input-validation framework should be introduced to prevent the intrusion vulnerabilities such as CSRF, XSS, SQL injection etc. Check for legitimate input data by validating the type, length, format, and range of data. The input data should be constrained to a list of values. Any characters that are not on the list can be rejected using regular expressions.

On-demand JavaScript is a technique used in many mashups. In On-demand JavaScript, script snippets are dynamically downloaded and interpreted on demand. These scripts can include malicious code aimed at exploiting security vulnerabilities such as XSS. This vulnerability can be prevented by ensuring that on-demand scripts are validated and the content generated from the scripts is encoded properly to prevent execution of malicious code.

Another solution is that before the server authenticates a new user, existing sessions should be aborted. Otherwise, intruders can intercept authenticated sessions and can capture the session details. These session details can then be used maliciously when a legitimate user creates a session with the same session details. These types of attacks can be prevented by either automatic logoff after a certain amount of time, or changing the session IDs with each request, or asking the user to reenter the user name and password whenever required.

An HTTP GET operation is assumed to be safe across domains, while HTTP POST is not safe. But to prevent XSS attacks, a server should avoid the use of HTTP GET requests to initiate changes; instead use only POST requests to initiate changes. Also HTTP POST access to any web site domain called on by a page other than the one from which the main page originated should be prevented. For example, a page from **http://xxx.org** can access

“safe” content from alternate domains like **http://xxx.edu** or **http://xxx.com**. But the page should not be allowed to post a form containing user input to anywhere except **http://xxx.org**. The XMLHttpRequest object provides a way to POST from script, and also prevents information from being posted to any domain other than the page domain. A technique for preventing CSRF attacks that mashup servers often face is to require each HTTP request to include a request-specific token to be transmitted to the server with each POST and GET request.

5 RELATED WORK

MashupOS [18] proposed by H.J. Helen Wang, a senior researcher at Microsoft Research defines protection and communication abstractions for the browser. MashupOS expands the trust model of SOP which matches the trust relationships found in a mashup. *ServiceInstance* and *CommRequest* are proposed for access-controlled content, and *<Sandbox>* and *<OpenSandbox>* for unauthorized content. MashupOS relies on SOP for cross-domain accesses, which has several vulnerabilities. Helen Wang has been working on solving the problem by providing a way for browsers to recognize code that comes from a third party, and to treat that code differently than that from the host website. She proposes enclosing third-party code in a "sandbox" tag. It would allow the larger website to make use of the code contained within the sandbox but treat that code as unauthorized content, with no authority outside the sandbox. Any information that the third-party code required could be included inside the sandbox. However, in order for this solution to be effective, the sandbox tag should be accepted as a Web standard. Wang has built a prototype of Internet Explorer that recognizes the above tag, but she notes that it would take time for the tag to be adopted in all browsers.

In contrast to MashupOS which requires different abstractions for expressing different types of trust relationships, Steven Crites proposed OMash [11], a simple abstraction that treats web pages as objects and allows objects to communicate only via their declared public interfaces. OMash does not rely on SOP for controlling cross-domain data exchange. Instead, each document is isolated apart from the public interface. This does not handle cross-origin content within the same document. Another security tool released by IBM called SMash [14], addresses the problem of securing mashup applications which mix active content from different trust domains. It comprises of central event communication hub and governed communication channels which mediate the communication between isolated components. These secure communication channels monitors information sent between tools, while still maintaining their separate identities and separate sets of permissions. However, once a piece of information has been communicated to another component, control over its use is lost.

Jackson and Wang proposed Subspace [19], a framework for secure cross-domain

communication between mashup components based on existing browser features. Each component is loaded in an *iframe* originating from a sub-domain of the integrator and communication is achieved by relaxing the domain attribute of the documents so that a communication object can be shared. A recent project by Google name Caja [14], also allows web applications of different trust domains to directly communicate with JavaScript function calls and reference passing. With the realization that a subset of JavaScript is an object-capability language, it is possible to translate scripts to this enforced subset and to grant these scripts only the privileges they require. It is therefore possible to isolate scripts from each other and from the global execution environment (i.e. the browser window) to the degree needed. As it is merely an enforced subset of JavaScript, Caja has the advantage of requiring no changes to any web standards. If a container wishes to allow an embedded application to use a particular web service, but not to send arbitrary network requests, then it can give the application an object that interacts with that web service, but deny access to XMLHttpRequest. Caja allows websites to safely embed DHTML web applications from third parties, and enables rich interaction between the embedding page and the embedded applications. Also a new policy SOMA (Same Origin Mutual Approval) was introduced by researchers of Carleton Computer Security Lab [13]. SOMA controls the unauthorized cross-domain information flows and prevents malicious servers from retrieving the page contents.

6 CONCLUSION

Web mashups blend data and functionality from multiple and often public sources. This introduces many new security risks. These new risks are of major concern when developing a mashup application. This paper focuses the problem of securing mashup applications which mix active content from different domains. As web mashups grow very rapidly, the problem of securing the information seems to be very important. Current browser security model, Same Origin Policy (SOP) is inefficient to support web mashup applications. SOP has several loopholes, which are discussed in this paper. Because of these loopholes, several vulnerabilities exist for web mashups. The main security vulnerabilities in web mashups are CSRF and XSS. In this paper, the existing solutions on mashup security are reviewed and suitable recommendations are provided.

REFERENCES

- [1] Programmable web, <http://programmableweb.com>, February (2008)
- [2] D. Merrill. Mashups: The new breed of web app. Technical report, IBM Corp., (2006)
- [3] Adam A. Nouredine, Meledath Damodaran.: Security in Web 2.0 Application Development. In: Proceedings of iiWAS, pp. 681-685 (2008)

- [4] Jonas Magazinius, Aslan Askarov, Andrei Sabelfeld.: A Lattice-based Approach to Mashup Security. In: Proceedings of ACM Conference on Computer and Communications Security (CCS 2010), pp. 15-23. April 13-16, Beijing, China (2010)
- [5] OpenAjax. Ajax and Mashup Security.
http://www.openajax.org/member/wiki/Ajax_Security_Resources
- [6] Cgsecurity.com. 2002. The Cross Site Scripting (XSS) FAQ. www.cgsecurity.com/articles/xss-faq.shtml
- [7] <http://webtrends.about.com/od/webmashups/a/webmashup.htm>
- [8] Computer Emergency Response Team (CERT) at Carnegie Mellon University:
<http://www.cert.org/advisories/CA-2000-02.html>
- [9] <http://twitspy.com>
- [10] http://en.wikipedia.org/wiki/Cross-site_scripting
- [11] Steven Crites, Francis Hsu, and Hao Chen: OMash: Enabling Secure Web Mashups via Object Abstractions. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), pp 99-107. October 27–31, 2008, Alexandria, Virginia, USA (2008)
- [12] <http://www-archive.mozilla.org/projects/security/same-origin.html>
- [13] Terri Oda, Glenn Wurster, P.C. van Oorschot, Anil Somayaji.: SOMA: Mutual Approval for Included Content in Web Pages. In: Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008), pp 89-98. October 27–31, 2008, Alexandria, Virginia, USA (2008)
- [14] F. D. Keukelaere, S. Bhola, M. Steiner, S. Chari, and S. Yoshihama.: Smash: secure component model for cross-domain mashups on unmodified browsers. In WWW '08: Proceeding of the 17th international conference on World Wide Web, pp 535–544. New York, NY, USA (2008)
- [15] C. Jackson and H. Wang.: Subspace: Secure cross-domain communication for web mashups. In: 16th International Conference on the World-Wide Web (2007)
- [16] http://en.wikipedia.org/wiki/DMZ_computing
- [17] Sara Javanmardi, Yasser Ganjisaffar, Cristina Lopes.: Scientific Mashups: The Issue of Trust in the Aggregation of Web 2.0 Content. In: Web Science Conf. 2010, pp 1-5. April 26- 27, Raleigh, NC, USA (2010)
- [18] H. J. Wang, X. Fan, J. Howell, and C. Jackson.: Protection and Communication Abstractions for Web Browsers in MashupOS. In: Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP 2007), pp 1–16. New York, NY, USA (October 2007)
- [19] Collin Jackson and Helen J. Wang.: Subspace: secure cross-domain communication for web mashups. In WWW '07: Proceedings of the 16th international conference on World Wide Web, pp 611- 620. New York, NY, USA (2007)
- [20] Adam Barth, Collin Jackson, John C. Mitchell.: Securing Frame Communication in Browsers. In: Proc. of the 17th USENIX Security Symposium. (USENIX Security 2008)

A Rule-Based Framework for Educational Board Games

Boyan Bontchev

Faculty of Mathematics and Informatics, Sofia University St Kl. Ohridski,
5 James Baurchier Blv., 1164 Sofia, Bulgaria
bbontchev@fmi.uni-sofia.bg

Abstract. In last decade, educational games such as quizzes, puzzles and quests have provoked an increasing interest in new methods of learning and appeared to be an appeal motivating students to study in a way rather different than the traditional one. Board games use rule-driven approaches for both presentation and control. Traditional quizzes, puzzles and quests can be presented as multimedia board games, often with artificial agents simulating the opponent or a collaborator. The paper presents in brief a formal model of positional board games serving for educational purposes and discusses description of rules for game execution in temporal context. The model is planned to be used as a basic framework for the development of a multimedia game platform for an e-learning system providing adaptive courseware. There are presented examples of positional games with education purposes and, as well, the basic architecture of a software game framework supporting the explained model.

Keywords: board games, education, model, framework.

1 Introduction

Modern technology-enhanced learning approaches rely on inclusion of multimedia content into traditional educational process, such as animations, video and games. Games are well-know mean to attract learner attention and to retain his/her interest to the lesson. In fact, they may bring a great entertaining, educative and cognitive value. In last two decades, development of different approaches complementing the traditional instructional learning design with educational games became a bold trend [1]. The educational games help organizing and provisioning of learning activities with predictable outcome, where goals, constraints, rules for payoff, and precisely defined consequences [2]. Such games have already been used successfully in school education as well as in universities and, as well, for professional training [3]. The added value brought by an educational game concludes in encouraging students to improve their knowledge by making them to solve various problems of particular scientific area or engineering practices and, thus, by improving their creative thinking [4].

Nowadays, quizzes, puzzles and problem solving staging are mostly used as educational game types [5, 6]. The author proposes problem-oriented board games to



be also included here, which supposes moving figures across a surface (board or a map) using counters or dices. Such approaches rely on profound knowledge of issues of course material for achieving safe navigation through a map or board. Board games may be played by single users alone or against an artificial agent presenting a real opponent [7]. Most often they are played by two or more players and enable participants to learn interactively from one another while playing for fun. There are designed educational games, which are combination between board game and quiz as they apply board rules for navigation within a quiz (to move from one question to another) and for control of the questions [8, 9].

The present paper is focussed on the management process of more general, multimedia-rich and problem-oriented quiz games, which are created by means of board game instruments. It sketches in brief a new, formalized model of such types of games already explained in [10]. This model is going to be used for assisting creation of a clearer specification of a software game framework and can be used for more precise analyze of performance of these type of games compared with other ones. It supports creation, management of internal logic and interpretation of results of quizzes, puzzles, quests, logical problems as well as instructional set of teaching activities presented as a race game. Finally, the model is going to be used as a basic paradigm for the development of a multimedia game for an e-learning system providing adaptive courseware.

2 Motivation

Until the moment, the best mean for realizing self-assessment tests or final exams is to develop quizzes. Quizzes, mazes and solution of specific problems are among most popular educational games, while board games are preferred most often for entertainment and strengthening intellectual potential. Usually, quizzes are utilized as tests for self- and/or official assessment, within the scope of given course or a specific topic. Usage of quizzes continues being within the frame of traditional quiz format, presentation and run time control. Though some quizzes may be developed as games and next played for fun, the majority of the existing tools provide mechanisms for construction of traditional quizzes. There are many examples of commercial tools for quiz design as presented in [10] such as free tools for creating and administering of quizzes used as online or desktop applications, and for provisioning automatic report of the test results. Some of them allow combination of test quiz with multimedia with special editors, for example for mathematical equations. There are several frameworks and tools to be used for design of educational quizzes. Some of them allow automatic generation of questions and, as well, automatic assessment of answers while others propose creation of quizzes by authoring tools [9, 11]. ELG [9] provides an authoring tool for personalization of each game depending learner's level of knowledge, preferences and educational goals. It uses rules which, depending on shown knowledge and defined goals, manage the transition from one game level to another. Other authoring tools offer opportunities to create parameterized questions [12]. QuizGuide and QuizJet [12] have been developed as adaptive systems helping

students to select self-assessment quizzes most relevant to individual needs and preferences of each learner.

Another class of games is based on the pedagogical approach in which students are actively involved with key concepts of a problem. Example of implementation of an active learning game is Genetic Algorithm Game [13]. This game is used in the area of engineering education and provides very positive results in student experience and conceptual learning.

Newly proposed map (or board) rule-driven approaches are a modern trend in quiz presentation and control. In these approaches profound knowledge of the taught course material assures safe navigation through a map (board) [8, 9]. The concepts of these type of games ensure some benefits such as players make take turns by rolling dices, questions may vary in terms of difficulty, and there could be applied various strategies for selection of question card for each player. Such games can be played by multiple users together, by a single user alone as a normal quiz and even by a single user against a simulated player (by an artificial agent).

Unlike quizzes, a board game is created by special game design tools and is based on a specific strategy for problem resolution in the scope of given configuration, usually, under given race conditions. Both quizzes and board games may be played by a single user and/or multiple users. More often, run time is divided into steps (passes, paces) which cannot exceed predefined time duration.

Usually, a quiz is composed by separate questions presented at run time in given time order. Statically built quizzes have a predefined flow of quiz questions, while dynamic quizzes select questions on run time in a random way or using given criteria. Within the quiz flow, each one of the questions represents a specific problem and requires one or several answers from a predefined answer set or needs to be answered by a predefined word or sentence (filling blank question type). There are other, more interesting types of questions, requiring sorting a list of words or solving a proper mapping between objects. Finally, there are questions of different type stating a general problem to be solved. Questions of the last type operate with graphical abstractions (figures) and the person solving such a quiz has to move, reorder or rotate them in order to solve the problem.

The author has developed a general quiz model [10], which could present all the quiz questions and processes controlling their answering. More, such a presentation is attractive, with high level of dynamism and appealing multimedia. On other side, the model is general enough to allow description of any situational problem and its solution suitable for e-learning purposes. A wide range of e-learning activities should have possible and not very complex model representation such as visiting a virtual museum and shooting pictures of some pieces of arts [14], collecting or selecting objects of given type residing specific locations, discovering a specific object on a map, finding (shortest) way within a labyrinth, etc. The execution of such game models should be straightforward and not very complex, following predefined game control rules.

3 Educational board game model

The goal of this section is to explain how a quiz can be modeled as a special board mini-game, with board of any form, positions, figures (objects), rules for manipulation and resulted effects. The model itself should be general enough in order to allow description and execution control of logical problems to be solved and, in general, of any learning activities and their workflow.

Generally, a quiz can be viewed on a non-restricted board, presented as an image map (contextual background specifying what should be done by the gamers) with given configuration of positions of various types over it. Normally, the capacity of each position is 1 but there are games with greater and even different capacities of positions. For grid games, configuration is defined by functions of neighborhood for all the positions. In non-grid board games, positions are spread over the background, however, even in such a case they may be viewed as aligned to an invisible grid as in graphical editors. Thus, it is possible to model dislocation of positions via grid coordinates which gives us the flexibility to move them to another grid cell when needed.

In board games, objects (figures) of different types (classes) may be allocated on some of the positions (or out of the map, on some reserved positions). The multiplicity (power) of each object type (let say – class) may be one (singleton) or greater than one. The gamer is supposed to do some actions over the objects such as filling text at some input fields, dragging and dropping an object from source to destination position, single or double mouse clicking over a figure or position, or even rotating a figure [15]. For a given object, each action may be executed only under some pre-defined rules (conditions) for the given object type. Such preconditions may concern the manipulated object itself (whether objects of its type could be moved, clicked, rotated, etc.) and/or the destination position and positions between the initial and destination one (for move, i.e. drag-and-drop actions). Such conditions may check whether the destination positions and positions between it and the source position are free, or other facts. Finally, after each allowed action, one or more resulted events may be fired depending on the action success. Together with these result events, the game execution engine should check if the final objective of the game is achieved. This check should be done at the end of each action in order to verify the finish conditions, as one of the resulted effects for the action. Thus, a quiz board game is defined as the following:

$$Game = \{GB, PT, P, OT, O, InitDisp, A, R, E, FC, Res\}$$

The model is defined formally in [10]. Below, there are outlined only the main building elements.

- GB – contextual game background image, may be visible or not
- PT = PT₁, PT₂, ..., PT_k - a set of all possible types of positions; the number of instances (i.e. of positions) of given type PT_p may be 1 or a integer number.
- P = P₁, P₂, ..., P_n - a set of positions (of equal or different types, k<=n) with predefined properties which may be checked by a checkProperty(P_i) and set by a setProperty(P_i) functions such as:

- type – $\forall \text{checkType}(P_i) \in \text{PT}$; position type is almost always fixed
- capacity – integer value (usually, $\text{checkCapacity}(P_i)=1$ and is constant), determines the maximum number of object instances which can be allocated on P_i
- title – a string value, empty by default, may be changed by the player (for editable titles)
- content – a string value, empty by default, may be changed by the player (for editable contents)
- visible – a Boolean value, true by default, may be changed by the player
- focused - a Boolean value, false by default, may be changed by the player by mouse click
- location – a 2-dimencional location of the position on the grid (the function $\text{checkLocation}(P_i)=(x, y)$, where x, y are integers, and usually are constant)
- $\text{OT} = \text{OT}_1, \text{OT}_2, \dots, \text{OT}_l$ - a set of all possible types of objects; the number of instances (i.e. of objects) of given type OT_p may be 1 or a integer number.
- $\text{O} = \text{O}_1, \text{O}_2, \dots, \text{O}_m$ - a set of objects (figures) of any type of the set OT with predefined properties which may be checked by a $\text{checkProperty}(\text{O}_i)$ and set by a $\text{setProperty}(\text{O}_i)$ functions:
 - type – $\forall \text{checkType}(\text{O}_i) \in \text{OT}$; object type is usually fixed but in some games (like chess) may be changed once (in chess, when a pawn reaches the final position) or more times under special circumstances; usually, there is one object for each object type: $\forall \text{OT}_p, \exists! \text{O}_q$.
 - title, content, visible, focused, location – like for the positions
- InitDisp – a two dimensional matrix of initial disposition of each object $\text{O}_i \in \text{O}$ over one (only one!) position $P_j \in P$. For every InitDisp , it is preserved that sum of all objects in given position is less or equal its capacity.
- $\text{T} = \text{T}_1, \text{T}_2, \dots, \text{T}_t, \dots, \text{T}_{\max}$ – a set of time discretets, or steps, when the moves are played; may be restricted or not; for each T_t there may be defined a maximum timeout or a delay.
- $\text{F} = \text{F}_1, \text{F}_2, \dots, \text{F}_c, \dots, \text{F}_f$ – a set of functions over P and O , such as:
 - $\text{containsTyped}(P_i, \text{OT}_k, \text{T}_t)$ – returns the number of objects of given type $\text{OT}_k \in \text{OT}$ at given step T_t (0 or an integer); $\text{contains}(P_i, \text{OT}_k, \text{T}_t) \leq \text{checkCapacity}(P_i)$;
 - $\text{containsAny}(P_i, \text{T}_t)$ – returns an array with numbers of objects of any possible type $\text{OT}_k \in \text{OT}$ at given step T_t (0 or an integer); $\text{contains}(P_i, \text{T}_t) \leq \text{checkCapacity}(P_i)$. For example, for the initial disposition given in table 1, we have $\text{contains}(P_i, \text{T}_0) = \{3, 0, \dots, 2, \dots, 4\}$;
 - $\text{get}(P_i, \text{O}_j, \text{T}_t)$ – removes object O_j from position P_i at time T_t ;
 - $\text{put}(P_i, \text{O}_j, \text{T}_t)$ – place object O_j at position P_i at time T_t ;
- $\text{A} = (\text{A}_1, \text{A}_2, \dots, \text{A}_s, \dots, \text{A}_a)$ – a set of player's actions, such as mouse click, double mouse click, and move (drag-and-drop). An object can be moved only from position to position.
- $\text{R} = (\text{R}_1, \text{R}_2, \dots, \text{R}_u, \dots, \text{R}_r)$ - a set of rules each under specific conditions oven F , for object type manipulation at specific position for given action; rules may differ from position to position and from action to action. For the drag-

and-drop (move) action from position P_{source} to P_{dest} for object of type O_i ($i=1..n$), the rule could be:

$$\text{Rule}(\text{Move}, P_{source}, P_{dest}, O_i, T_t) = \{ \text{if containsTyped}(P_{source}, OT_k, T_t) \geq 1 \text{ and containsAny}(P_{dest}, T_t) < \text{checkCapacity}(P_{dest}) \}$$

- e.g., for P_{dest} admissibility for allocation object of type OT_k over P_{dest} is checked, etc.), for the time step T_t . There may be defined some restrictive rules forbidding moving an object from P_{source} to P_{dest} , providing certain conditions are met. to As well, for the action click $P_{source} \equiv P_{dest}$

- $E = (E_1, E_2, \dots, E_f, \dots, E_c)$ – a set of effects to be applied after each action over object of specific type residing given position: $E_f = \text{Effect}(\text{Action}, P_{source}, P_{dest}, OT_i)$ may be a specific multimedia effect, possible increment of a game result, check of a position/object property, etc.
- FC – final condition – for example, arithmetic expression over F and attributes of P and O, check for the game result, etc. If the final condition returns true, the game is over.
- Res – game result as an arithmetic condition over F and attributes of P and O.

The model is flexible and highly extendible and provides a powerful paradigm for facile and rapid construction of rich multimedia games such as quizzes, quests, puzzles and even instructional sets of e-learning activities, in a uniform way. Next, they will be presented examples of quizzes and problem-oriented games, which are described as educational board games according this model.

4 Sample educational board games

As stated before, quizzes represent the most popular educational game type. Within a quiz, each mini-game will present a given quiz question. Unlike traditional quiz games, here the questions will appear in an appealing way, with advanced multimedia. Moreover, such board mini-games can represent not only quizzes but also specific logical problems to be solved. As well, they may present any set of teaching activities organizes in a list or set with random choice.

Questions with one exact answer, e.g. “How many are the months in a year?” may be presented by a single position with title equal to the question and empty content editable through mouse click; after clicking over the position, an entry field will appear and the player will enter the answer. Next, as an effect from the entering the answer (as position content), the position content will be compared to the right answer. The final condition will check if the position content is not empty. Questions with answers of the types ‘one of many’, ‘many of many’, and ‘filling blank’ have a straightforward and easy presentation. For such types of questions, answers are going to be represented as textual or multimedia objects and the player have either to select them by clicking or to move one or many of them to the position of right answers.

Figure 1 shows a quiz problem where players have to sort fishes as freshwater fish species and carnivores by moving them to one of two aquariums. Here, the mapping between positions (aquariums) and objects (fishes) is of type N to M (in fact, 2 to 7). The game rules should prohibit dropping animals at other position than the aquariums. Thus, as far as the objects may be dropped only at one of the aquariums (and possibly disappear after dropping within an aquarium), the condition for game end simply checks whether there are no animals at the starting positions and the position of each aquariums contains only the right fishes.

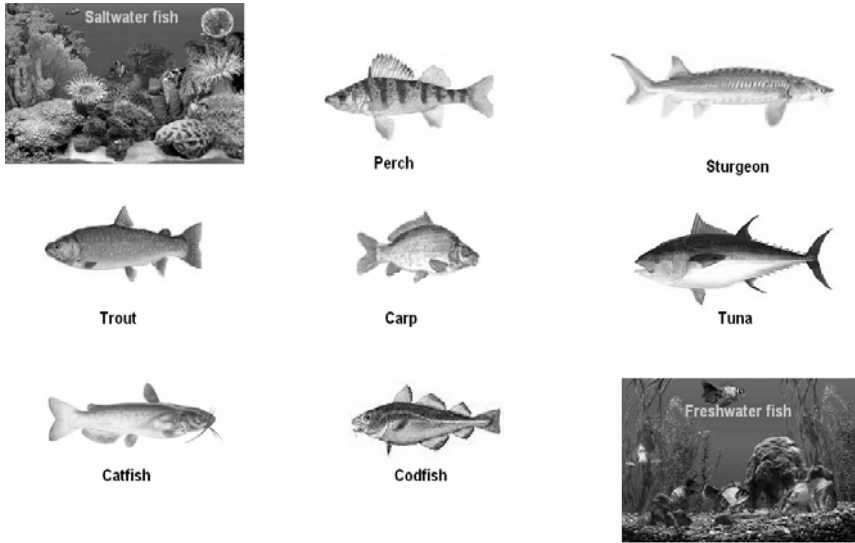


Fig. 1. Initial view of the fish species quiz game.

The board game model follows below:

- PT = Saltwater, Freshwater
- P = P₁, P₂, P₃, P₄, P₅, P₆, P₇, P₈, P₉;
 setType(P₁, Saltwater), setType(P₂, Freshwater), set P₃, P₄, P₅, P₆, P₇, P₈, P₉
 as ServicePosition's and invisible (used only for service purposes); all the
 capacities are equal to one.
- OT = Perch, Sturgeon, Trout, Carp, Tuna, Catfish, Codfish
- O = O₁, O₂, O₃, O₄, O₅, O₆, O₇,
 setType(O₁, Perch), setType(O₂, Sturgeon), setType(O₃, Trout),
 setType(O₄, Carp), setType(O₅, Tuna), setType(O₆, Catfish),
- setType(O₇, Codfish),
- InitDisp – shown in table 1 below.

Table 1. Initial disposition matrix.

Object\Position	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇	P ₈	P ₉
O ₁			1						
O ₂				1					
O ₃					1				
O ₄						1			
O ₅							1		
O ₆								1	
O ₇									1

- $R = R_i(\text{move}, P_i, P_{ii}, O_j)$, for $i=3..9$, $ii=1..2$, $j=1..7$, apply {get($P_i, O_j, 1$); put($P_{ii}, O_j, 1$), play(E_1);}
- $E = E_1=\text{CorectMove.wav}$, $E_2=\text{WellDone.wav}$, $E_3=\text{Bad.wav}$
- $FC = \{\text{containsAny}(P_i) = \text{empty}\}$, for $i=3..9$
- $Res = \{\text{if} (\text{containsTyped}(P_1, O_1)==1 \text{ and } \text{containsTyped}(P_1, O_2)==1 \text{ and } \text{containsTyped}(P_1, O_3)==1 \text{ and } \text{containsTyped}(P_1, O_4)==1 \text{ and } \text{containsTyped}(P_2, O_5)==1 \text{ and } \text{containsTyped}(P_1, O_6)==1 \text{ and } \text{containsTyped}(P_2, O_7)==1) \text{ then } \{\text{result}=\text{true}; \text{play}(E_2);\} \text{ else } \{\text{result}=\text{false}; \text{play}(E_3);\} \}$ – only the tuna and codfish live in saltwater.

Classical image or word puzzles have easy construction process as board mini-games. In [10] there is presented a positional quest board game for matching symbols with minimum number of mouse clicks, for shortest time possible. Thus, the model is able to support games with timing constraints, where the context may depend on the time when the action is entered (e.g., par and even step number). Here, there is given a model of a logical content-based game – the hangman game. For each game play, a word is chosen from a dictionary, and the game shows its characters as ‘-’. The only possible action is mouse click on a symbol from the alphabet, where each click makes the selected symbol to disappear from the alphabet and does a check whether the selected symbol is available within the word. If yes, then (in order not to be chosen again) and replaces the ‘-’ symbol at all position where it participates within the words (fig. 2); if not, next part of the gallows is shown. If all the parts of the gallows are visible, the game is lost. The player finishes successfully if the other word symbols are guessed (for the word in fig. 2, these are ‘G’, ‘V’, ‘Z’ and ‘R’).



Fig. 2. Intermediate view of the hangman game with five characters guessed.

The model of the hangman model follows bellow:

- PT = ‘_’, ‘|’, ‘O’, ‘/', ‘\’ - position types for the gallows; ‘-’ – type of symbols not yet guessed; ‘A’, ‘B’, ‘C’, ..., ‘Z’ - alphabet letters.
- P = P₁ - P₁₃ - for the gallows:
 setType(P₁, ‘_’), setType(P₂, ‘|’), setType(P₃, ‘|’), setType(P₄, ‘_’),
 setType(P₅, ‘_’), setType(P₇, ‘|’), setType(P₈, ‘O’), setType(P₉, ‘|’),
 setType(P₁₀, ‘/’), setType(P₁₁, ‘\’), setType(P₁₂, ‘/’), setType(P₁₃, ‘\’);
 for P₁₄ - P₂₅ - for the word, with maximal length 12 characters:
 setType(P_i, ‘-’), i=14, ..., 25 (all the characters are unknown initially);
 for P₂₆ - P₅₂ - for the alphabet letters: setType(P₂₆, ‘A’), ..., setType(P₅₂, ‘Z’);
- OT = ‘A’, ‘B’, ‘C’, ..., ‘Z’ - alphabet letters
- O = ∅ – the objects will be put within the InitDisp
- InitDisp = {

```

int gallowsPointer=0; string word=getRandomWord();
forall(int i=0, i<lenght(word),i++) {
    put(P14+i, charAt(word, i), 1);
    setVisible(P14+i, charAt(word, i), false);
}

```

– disposes the symbols of the chosen word and makes them invisible; in such a way, the player will see ‘-----’ instead of ‘hello’. Note that the function getRandomWord() should be coded apart.
- T = ∅ – the game is to be played with no time constraints
- A = mouseClicked
- E = E₁=Splash.wav, E₂=WellDone.wav, E₃=Wrong.wav
- R = {

```

R1(mouseClick, Pj=26..52) = {
    forall(int i=0, i<lenght(word), i++) {
        if(checkType(Pj)==checkType(Pi+14, any)) then {
            setVisible(Pi+14, any, true); setVisible(Pj, false);
        }
        if(checkVisible(Pj)==true) then {
            setVisible(P1+gallowsPointer, true); gallowsPointer++;
            play(E3)
        }
        else play(E2);
    }
}

```
- FC = { if(gallowsPointer==13) return true;
 forall(i=0;i++, lenght(word)) {

```

if(isVisible(Pi+14, any)==false)
return false;
}

```

 return true;
 }
- Res = {if(gallowsPointer==13) then false else true;}

5 Software architecture of a board game framework

This model of quiz and puzzle games provides a powerful paradigm for facile and rapid construction of rich multimedia games of such types in a uniform way. It is going to be used as a base for construction of a software framework for both design and play of educational board games. The complexity of such a game will depend on the complexity of the rules and conditions and checking functions used for rule definition. Moreover, even more complex position board games as chess and backgammon can be defined by the model sketched over via function of neighborhood; however, they are not appropriate for educational purposes.

The software framework will include a graphical designer of games of such types with various levels of multimedia enrichment and complexity and, as well, game play controller. Figure 3 represents the basic architecture of this framework. The graphical designer provides authors with means for definition the background, places and objects images, initial disposition, actions, events, final condition, result and, most important – the control rules.

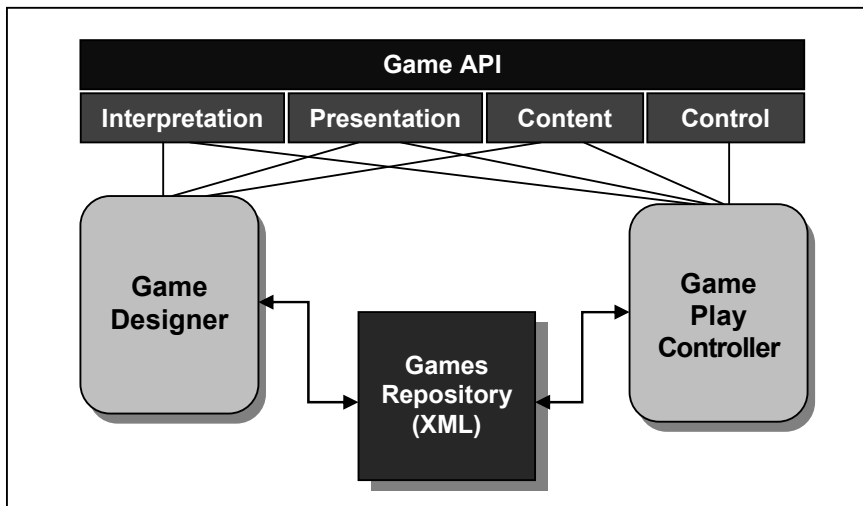


Fig. 3. Principal architecture of the educational board game framework.

The game definition is saved as XML document together with the images and next may be edited and/or executed by the play controller. Both the game designer and play controller use common API for interpretation (reading game documents, rule validation, etc.), presentation (text markup), content management (i.e., for choosing a word from a dictionary) and game control (for rule execution). Control API is used only by the game play controller as far as this component is responsible for game execution.

The educational board game framework is going to be used for development of various game platforms.

6 Conclusions

The paper has presented the idea of modeling quizzes, quest, puzzles and logical problems as multimedia board games, and how such model may be used for creation of a software game framework for board game design and execution. The model is quite powerful and is able to describe also other set of instructional activities suitable for e-learning. Using this model, various mini-games could be easily described formally and, next, practically developed provided there is available a game framework for building such board mini-games. In fact, it allows presentation of more complex logical problems which have to be solved by several actions delivered by the player according some rules and context conditions. Such resolution of logical problems is suitable for active online learning. Finally, a complex set of instructions and activities within given domain context can be represented as a board game using the same model. Examples for such scenarios are activities typical for visiting virtual museum, following a receipt for preparing a dish, locating and discovering an object using geographical map, and many more.

The game framework will provide a game designer providing facilities for easy construction of multimedia board games. The examples of such games given over and in [10] should be designed by non-programmer authors without any great effort. The XML internal representation of the designed games will facilitate interoperability with other gaming platforms. Next, the games will be played under control of the game control engine executing the rules and the effects for each player's action under certain conditions according the model prescriptions.

The future work will continue with designing a functional interface for game rule creation and, therefore, creation of an online tool for authoring and generation of Flex board games compliant with the proposed model. By means of game metadata descriptions, such games will be incorporated within narrative storyboard and will be delivered to learners with appropriate profile (learning style, preferences, etc.). At next stages, usage of artificial intelligence agents is possible - as virtual opponent of the player or as virtual adviser of the gamer helping him/her in finding the right solution of the stated problem.

Acknowledgments

The work reported in this paper is supported by the ADOPTA project funded by the Bulgarian National Science Fund under agreement no. D002/155.

References

1. Dempsey, J. et al: Instructional applications of computer games", American Educational Research Association, New York, 1996, pp. 8–12 (1996)

2. Siang, A., Rao, R.: Theories of learning: A computer game perspective. Proc. of the IEEE Fifth International Symposium on Multimedia Software Engineering (ISMSE'03), pp. 239-245 (2003)
3. Prensky, M.: Don't bother me, mom, I'm learning!, St. Paul, Minn., MN: Paragon House (2006)
4. Salen, K., Zimmerman, E: Rules of play: Game design fundamentals. MIT Press, Cambridge, MA, USA, October (2003)
5. Batson, L., Feinberg, S.: Game Designs that Enhance Motivation and Learning for Teenagers, Electronic Journal for the Integration of Technology in Education, Vol. 5, pp. 34-43 (2006)
6. Ferreira, A. et al.: The common sense-based educational quiz game framework "What is it?", ACM International Conference Proceeding Series. Vol. 378, pp. 338-339 (2008)
7. Bontchev, B., Gabarev, N., Pavlov, H.: A Mobile Chess Game, Proc. of 16th SAER Conference, Varna, Sept. 2002, pp. 138-143 (2002)
8. Feng, K.: Joyce: A Multi-Player Game on One-on-one Digital Classroom Environment for Practicing Fractions, Proc. of the Fifth IEEE Int. Conf. on Advanced Learning Technologies (ICALT'05), pp. 543-544 (2005)
9. Retalis, S.: Creating Adaptive e-Learning Board Games for School Settings Using the ELG Environment, J. of Universal Computer Science, vol. 14, no. 17 (2008), pp. 2897-2908 (2008)
10. Bontchev, B., Vassileva, D.: Educational Quiz Board Games for Adaptive E-Learning, To appear in the Proc. of Int. Conf. ICTE'2010, Paris, France, 28-30 June, (2010)
11. Dalziel, J.: Using LAMS Version 2 for a game-based Learning Design, Special issue on Comparing Educational Modelling Languages on the "Planet Game" Case Study, JIME, November (2008)
12. Hsiao, I-Han, S. Sosnovsky, Brusilovsky, P.: Adaptive Navigation Support for Parameterized Questions in Object-Oriented Programming, LNCS, Vol. 5794/2009, ISBN 978-3-642-04635-3, pp. 88-98 (2009)
13. Weck, O., Kim, I. Y., Hassan, R.: Active Learning Games, The 1st CDIO Annual Conference, 06-09 June 2005, Kingston, Ontario. pp 1-15 (available at www.cdio.org) (2005)
14. White, M. et al: ARCO - an architecture for digitization, management and presentation of virtual exhibitions, Proc. of Computer Graphics Int. Conf., pp. 622-625 (2004)
15. Bontchev, B., Vassileva, D.: Modelling educational quizzes as board games, Proc. of IADIS e-Society 2010 Conf., ISBN: 978-972-8939-07-6, Porto, March, pp.20-26 (2010)

Regular Sparsity Map

Ina Naydenova

“St. Kliment Ohridski” University of Sofia, Faculty of Mathematics and Informatics
Sofia 1164, Bulgaria
ina@fmi.uni-sofia.bg

Abstract. Computing multidimensional aggregates is a performance bottleneck for many OLAP applications. Sparse data tend towards the data explosion problem in precomputation process and decreases the performance of OLAP. Form another point of view there is useful information that sparsity can give us. In this paper we introduce a new object – “regular sparsity map”. The map saves information about specific empty domains of multidimensional cubes and provides business analysts with the ability to define rules and place data constraints over the multidimensional cube. A formal definition and a short overview of the possible map application are presented.

Keywords: dimension, cube, OLAP, sparsity, regular, random.

1 Introduction

Data Warehouse (DW) and Online Analytical Processing (OLAP) are essential elements for many decision support and business intelligence (BI) systems. They are complementary technologies - a DW stores and manages data, while OLAP transforms data into possibly strategic information. OLAP systems use a multidimensional to organize their data. In [7] we consider some difficulties related to the user requirements and the high-dimensionality of the requested reports. We discuss several approaches of how to extend the multidimensional model in order to avoid the encountered problems. Here, we present a further elaboration of the idea of a “map of feasible tuples” (named “regular sparsity map” in the current paper). The regular sparsity map object allows improvements in the technical OLAP implementation (concerning performance and storage issues) and enables some extra facilities for the OLAP end users.

In this paper we introduce a new classification of multidimensional cube sparsity phenomenon, define the “regular sparsity map” object and investigate the map applicability in three directions – data quality, performance issues and end user interface facilities.



2 The Multidimensional Data Model

DW usually requires consolidation of data from many heterogeneous sources. The different sources might contain data of varying quality, or use inconsistent representations, codes and formats, which have to be reconciled. Since data warehouses contain consolidated data, perhaps from several operational databases, over potentially long periods of time, they tend to be orders of magnitude larger than operational databases. In data warehouses, historical and summarized data is more important than the detailed one. A popular conceptual model that influences the front-end tools, database design, and the query engines for OLAP is the multidimensional data view.

A general multidimensional model includes two widely recognized entities: the fact and the dimension. The fact is the subject of decision-oriented analysis. It usually consists of the measures, metrics or facts of a business process, and is represented by means of a data cube. Dimensions correspond to a perspective, under which facts can be meaningfully grouped and analyzed [5]. The elements of a dimension could be aggregated using hierarchy. For instance, “May 2008” could be summarized into “Second Quarter 2008” which in turn would be summarized in the “Year 2008”.

Regardless of common basis of the model there are a lot of formal definitions. One can find a multidimensional model specified as a set of dimensions, measures and data cubes with corresponding definitions of object schemas, domains and applicable operators in [9] or using a multidimensional space (Cartesian product of relational attribute projections) structured by the generalization order between tuples (cube lattice as a convex space is considered) in [1]. Since even tabular data, such as relations, can be thought of as multidimensional, many definitions incorporate this point of view: multidimensional database algebra, specified over table schemas, is defined in [3] and [12].

Throughout this paper we assume a simplified conceptual cube model that treats data in the form of n-dimensional cubes. Cube models have a more or less explicit notion of fact, measure and dimension. The hierarchy between the various levels of aggregation in dimensions is not explicitly captured by the schema.

2.1. The Model Definition

- *Dimension* is a non-empty finite set;
- *Multidimensional space* over dimensions D_1, D_2, \dots, D_n ($n \geq 1$) is the Cartesian product $S = D_1 \times D_2 \times \dots \times D_n$. It contains n-tuples (x_1, x_2, \dots, x_n) where $x_1 \in D_1, x_2 \in D_2, \dots, x_n \in D_n$
- *Rectangular domain* in multidimensional space S is a subset $M \subseteq S$,
 $M = A_1 \times A_2 \times \dots \times A_n$, where $A_1 \subseteq D_1$, $A_2 \subseteq D_2$,
 $\dots, A_n \subseteq D_n$;

- \emptyset is a special value named “empty value”;
- *Fact F* is a set, where $\emptyset \in F$;
- *Cube* is a function $C: S \rightarrow F$, where S is a multidimensional space, F is a fact;
- *Cell* in the cube $C: S \rightarrow F$ is a pair $c = (t, f)$, where $t \in S$, $C(t) = f$. The cell is *empty* if $f = \emptyset$ and *non-empty* otherwise;
- *Set of empty cells* in the cube $C: S \rightarrow F$ is the set $E(C) = \{t \in S \mid C(t) = \emptyset\}$, $E(C) \subseteq S$

2.1. Sparsity Definition

Sparsity refers to the natural phenomenon evident in multidimensional data: not all of the cells in the logical cube will ever contain data. The more empty cells found in a cube, the sparser the cube data is. This is measured by the density coefficient.

$$\text{Density coefficient of cube } C: S \rightarrow F \text{ is a ratio } c = \frac{|S| - |E(C)|}{|S|}$$

If we have 60 stores, 500 products, 70 000 customers and 12 months in a year, our cube has a potential $60 \times 500 \times 70000 \times 12 = 25\,200\,000\,000$ cells, but we might only have 360 000 000 non-empty cells in our measure (40 000 customers shopping 12 months a year, buying average on 25 products at 30 stores) making our cube 1.42% dense.

To provide a practical baseline expectation for sparsity in [15] the authors examine data sparsity in a variety of models with a sample of seven companies. Each company has a variety of models (e.g.g, P&L, balance sheet, cash flow, sales analysis, HR/labor analysis, budgeting and forecasting, industry-specific models, etc.) with differing dimensions. The industry-specific models include insurance claim analysis, telco call analysis and revenue-per-user-analysis and a medical device company’s sales analysis. The research shows that: (1) data density in all cases is significantly less than 1 percent – i.e., extremely sparse; (2) as the number of dimensions increases, so does the sparsity of the data (models reviewed have between 5 and 16 dimensions); (3) extreme sparsity exists in all industry-specific models (all models have density of less than 1 billionth of a percent).

Because the extremely sparse cubes are frequent phenomenon, OLAP engines offer different methods of increasing the performance and reducing the size of sparse cubes: concatenated indexes (composites), compression techniques, cube partitioning, sparsity and aggregate advisors. But all of these methods do not take account of the nature of sparsity and did not divide the sparsity into any types.

2.2. Sparsity classification

A closer scrutiny reveals that there could be some difference between empty cells in terms of the causes provoking the cell’s emptiness. We could not found proper classification of sparsity types neither in the literature nor in the current data

warehouse and OLAP researches. Actually, we found only two sparsity classifications based on the empty cells distribution. Oracle OLAP divided the sparsity into controlled and random: controlled sparsity means that a range of values of one or more dimensions has no data, random sparsity means that null values are scattered throughout a cube [11]. In [4] sparsity patterns are classified as random, stripe, cluster and slice type in relation to their shape.

Our experience in sparsity problems points at another kind of classification, so we introduce a new one - we distinguish two types of cube's sparsity:

- *random* sparsity;
- *regular* sparsity;

If one cell is empty because of the semantics of the modelled business area (the semantics enforces lack of value), then we witness "regular sparsity". If the cell is empty, but it is possible it had a value, "random sparsity" is what we have. Missing data in random sparsity usually express zero values, whereas the regular sparsity expresses inapplicable values. In [7] we point out several forms of regular sparsity (irrelevant dimensions, segmentation of dimensions, dimension changes over time).

3. Regular Sparsity Map Definition

To formally distinguish regular from random sparsity, we introduce the following definition:

Regular sparsity map (RSM) of the cube $C: S \rightarrow F$ is the set $R_c \subseteq E(C) \subseteq S$.

A regular sparsity map (or shortly map) RC determines the cells which are empty because of regular sparsity (business rules, formal requirements, natural dependencies, etc.).

The set difference $E(C) \setminus R_c$ determines the cells which are empty because of random sparsity.

Map Representation

One of the main characteristics of regular sparsity is the knowledge of its existence in advance. The map can be constructed over the multidimensional space S before the construction of cube C . So the construction of the map can be an important step in the process of a multidimensional model design. A convenient approach is that the regular sparsity map is constructed on the basis of a set of business rules. The first step is defining the rules and specifying the relationships between them. Then every business rule has to be transformed into a combination of rectangular domains. The final step is setting a Boolean algebra expression that represents the map using rectangular domains connected by union or intersection operations.

Examples of map representation

The business rule in the example above states: “Every product sold in store “Boyana” is on the list Z”.

Here is another example of business rules that does not depend of the cube: We have a multidimensional space S over dimensions “Time” (D_1), “Services” (D_2), “Sales channels” (D_3), “Regions” (D_4), “Client types” ($D_5 = \{‘private person’, ‘organization’\}$), “Gender” ($D_6 = \{‘Male’, ‘Female’, ‘N/A’\}$) and “Branch of business” (D_7):

- The business rule “Organizations do not have a gender” can be expressed by the regular sparsity map of arbitrary cube C in multidimensional space S: $Rc = \{(d_1, d_2, d_3, \dots, d_7) \in S \mid d_5 = \text{“organization”}, d_6 \neq \text{“N/A”}\}$ (table 1).

Travel Insurance		Number of clients			
Service	By agent	Private person		Organization	
Sale Channel	Region	Female	Male	Female	Male
Client Type	Sofia				
Gender					
	Jan 2008	28 985	29 163		
	Feb 2008	28 123	29 400		
	Mar 2008	27 200	29 841		

Car Insurance		Number of clients			
Service	By agent	Private person		Organization	
Sale Channel	Region	Female	Male	Female	Male
Client Type	Sofia				
Gender					
	Jan 2008	43 734	12 167		
	Feb 2008	44 376	12 152		
	Mar 2008	44 712	12 442		

Table 1. Empty rectangle described by regular sparsity map

- The business rule “There are no travel insurance offices in the cities of Stara Zagora, Varna and Burgas” can be expressed by the regular sparsity map of arbitrary cube C in multidimensional space S: $Rc = \{(d_1, d_2, d_3, \dots, d_7) \in S \mid d_2 = \text{“Travel insurance”}, d_4 \in (\text{“Stara Zagora”}, \text{“Varna”}, \text{“Burgas”})\}$.

4. Map Applicability

According to the independent OLAP Survey conducted by Nigel Pendse [14] the top three problems in the field of Business Intelligence (BI) over the past years are slow query performance, poor quality data, and company politics. The knowledge preserved in a regular sparsity map can be used in the solutions of two of these problems: the improvement of data quality and query performance. In addition, it enables the implementation of some user interface facilities.

4.1. Data cleaning

In the business world, incorrect data can be costly. The dirty data in the warehouse systems often is an obstacle for the practical application of BI systems. Usually, only after the data is loaded and the first results are obtained it is clear that there are defects in the data. The discovering and elimination of these defects is quite hard procedure because the input data is related with many dependencies and passes through a number of transformations until it is presented to the multidimensional model. A second time data loading is often necessary and an execution of all the steps over again. The solution of this problem is the data to be verified on a possibly earlier stage of its processing. The regular sparsity map describes part of constraints over the data in the terms of multidimensional model, which is close to the concepts of the business analysts. At the same time it enables easy implementation of automatic data tests before receiving the results by the end users.

The development of a module for business constraints and dependence enforcement (RSM error module) is a direct application of the map information. The module can have several key functionalities:

- Validation of the regular sparsity map definition over a trusted data cube (cube without dirty data); this functionality is to be used immediately after the process of map construction, in order to check the correctness of the specified constraints;
- Errors detection and correction; after the validation of map definition, the constraints can be enforced over unverified data:

- o When data is loaded into a database, the RSM error module has to detect inconsistent situations and report them. The ability for synchronization of the data and map definition will facilitate the process of data cleaning. The synchronization process can nullify the data according to the map constraints or modify the map definition in conformity with the loaded data. Because there might be some data exception from the general business rule (for example, product X is not available in store Y, but because of the Christmas promotion X was available in December 2007) it would be better if the synchronization can be made partially (in portion) regarding specific constraints.

- o The map information can be used for automatic generation of database constraints over the source data. Constraint generation is acceptable if we use a relational OLAP storage model (ROLAP), but the analogous type of constraints can be implemented in multidimensional OLAP storage model (MOLAP), too (provided the multidimensional database system supports such);

- Automatic generation of skeleton (draft) regular sparsity map definition from the cube data by means of data mining techniques.

Let us have a cube C varying over 6 dimensions: Time, Service, Sales Channel, Regions, Client types, and Branch of business. We have defined a RSM over C that specifies the following rules:

1. If a service is “travel insurance” and sales channel is “by agent” then regions are Sofia, Pleven, Plovdiv and Ruse.
2. If a service is “credit cards” the sales channel are “internet sales” and “direct sales”
3. If client type is organization then gender is unknown.

4. If client type is private person then branch of business is unknown.

When the snapshot for February 2008 was loaded in the data warehouse system, the gender of all “Ltd” corporate clients loaded from a data source SYS1 is set to Male (the SYS1 is a new source system and there is code matching misunderstanding). The error module reports that the rule 3 is violated. The module also reports that there is a sales channel that violates rule 2. The further investigation shows that a new promotional credit card program is started but the channel is not entered in corresponding data warehouse table.

A classification of the major data quality problems is presented in [16]. According to this classification the RSM can support the process of Instance Level inconsistent data detection. The authors discuss that in order to detect which kinds of errors and inconsistencies are to be removed a detailed data analysis is required. In addition to a manual inspection of the data or data samples, analysis programs should be used to gain metadata about the data properties. The constraints defined with a regular sparsity map are an additional source of metadata. In addition, RSM based validation has the advantage that every modification in the RSM constraints immediately will be taken into account during the data cleaning process.

RSM advantages: Business rules that imply cube sparsity are collected in a common object with automated error reporting capabilities, instead of specific custom solutions. They are a source of a missing values metadata which kind of information usually is not enough in data warehouse systems (see [8]).

4.2. Query performance and storage consideration

4.2.1. Composite selection

Some of the most important techniques for queries performance improvement in OLAP applications are the composites. A composite is a concatenation of several dimensions, so it contains one tuple for each set of dimension values that identifies non-empty cells in the cube. The choice of what dimension is concatenated in a composite and how many composites are created is not an easy task. The effective composite is a small one, i.e. with less meaningful dimension value combinations. If we use a relational storage, the correlation between dimensions can be a reason for a set of dimensions to be stored together in joint relational scheme, instead of separate dimension schemes (see [17]). This technique is also referred to as "flattening dimensions" (see [2]). OLAP designers take a decision which part of the dimensions to form a composite on the ground of their knowledge about the business area and inner object dependences. They can also use sparsity wizards or manually analyze the data in fact tables. The RSM can support the process of composite candidates' evaluation.

Let we have a multidimensional space S over dimensions “Time” (D1), “Services” (D2), “Regions” (D3), “Client types” (D4 = {‘private person’, ‘organization’}), “Gender” (D5 = {‘Male’, ‘Female’, ‘N/A’}), a fact F and a cube $C: S \rightarrow F$

If we use a relational database approach of cube implementation, the storage allocated by the cube C can be evaluated by the following formula:

$$CZ = \sum_{j=1}^n (Size(F) + \sum_{i=1}^5 Size(D_i)),$$

where $Size(D_i)$ is the size of data type for a dimension D_i identifiers, $Size(F)$ is the sum of sizes of data types for a set of measures F and n is the number of non-empty cube cells.

The possible combinations between dimensions “Client types” and “Gender” are $|D4|*|D5| = 6$. If we have a rule “Organizations do not have a gender” expressed by the RSM, the set of meaningful combination can be reduced to the number of 4 (the ratio between meaningful and all possible combination is promising - 33% meaningless combinations). So the concatenation of dimensions “Client types” and “Gender” is a good designer solution. In this case we have a multidimensional space S' over dimensions “Time” (D'_1), “Services” (D'_2), “Regions” (D'_3), “Composite1” ($D'_4 = \{ 'private person, male', 'private person, female', 'private person, N/A', 'organization, N/A' \}$), a fact F and a cube $C': S' \rightarrow F$.

The storage allocated by the cube C' can be evaluated by the following formula:

$$CZ' = \sum_{j=1}^n (Size(F) + \sum_{i=1}^4 Size(D'_i)),$$

where $Size(D'_i)$ is the size of data type for a dimension D'_i identifiers, $Size(F')$ is the sum of sizes of data types for a set of measures F' and n is the number of non-empty cube cells.

The storage size in the second approach $CZ' < CZ$ because the number of non-empty cells n is the same for C and C' and $Size(D'_4) < Size(D_4) + Size(D_5)$.

The query performance depends on the RDBMS optimization techniques but in average it would be the same or faster (in general, the number of join operations is smaller).

RSM advantages: The sparsity wizards or manual analysis of meaningful combinations are made on the basis of current data snapshots and therefore the information is not accurate enough. The RSM is independent of the data extract. Besides, the RSM enables composite selection to be done in the very stage of database design, even when a real-life data is not available.

RSM disadvantages: It does not take into account a random sparsity that also plays a part in a process of meaningful combinations evaluation.

2.2. Data compression

We observed a problem related to the usage of measures which are non-additive (cannot be computed by the cells of the lower hierarchy level). In the case there is a method for calculation of aggregated measure values, but this method is not supported by the OLAP engine (or for some other reasons is out of the multidimensional database). So, one must load the calculated values in the multidimensional database that can be unacceptable in terms of storage magnitude. The problem could not be resolved by means of dynamic aggregation, because all non-additive data must be preaggregated.

We developed a data storage reducing scheme to solve the described problem. Firstly, we detect additive sections in non-additive hierarchy structures, in particular, elements with only one descendant value (called “phoney nodes”) following the

hierarchy. Then we describe a physical model which calculates the collected additive sections online. A regular sparsity map can be useful in this process. In the process of phoney nodes detection we use structures that are in fact a specialized implementation of the regular sparsity map. In [6] we report the result of the method over 6 cubes with real-life data for a particular month. The percent of the cube size decrease (storage gain) varies between 32% and 96% depending on the data distribution. The average storage decrease for all cubes is about 48%.

RSM advantages: The existing data compression solutions work mainly on physical level and do not take account of the nature of sparsity. With the information preserved in RSM the proposed reducing scheme can be implemented as part of the OLAP engines. Our implementation as a logical layer over the Express OLAP engine proves its efficiency.

4.3. Cube partitioning

Partitioning is a method of physically storing the measures in a cube. This technique improves performance as it keeps the working set of data smaller both for queries and maintenance, enables parallel cube processing etc. Each partition functions like a smaller cube. The regular sparsity map is a good basis for implementing software tools that automate the cube partitioning process. Development of partitioning algorithm that breaks down the cube to sub-cubes, many of which are completely empty (according to the regular sparsity map contents) could improve the query performance. In [10] we found a synthesis and decomposition algorithm for producing normalised OLAP cubes based on the functional dependency between dimensions (structural sparsity). The regular sparsity map can also be used for OLAP cube normalization. It not only stores functional dependences but also dependences between subsets of dimension values.

Example: In a real-life system we have designed a cube varying over 15 dimensions: time, services, sales channels, regions, client types, gender, branch of business, number of the client contracts (for example “1 contract”, “2 contracts”, “over 2 contracts”), classes of client term (“3 months”, “6 months”, “long-term client”), state of client (“new”, “restored”, “non-active”) and others. If the number of services and regions elements is more than 2000, many of MOLAP engines will have data explosion performance problems (see [13]). We overcame the difficulties but this was related with additional work. The main strategy of our approach was to break down sparse cubes to smaller compact sub-cubes. Instead of the described cube we design derived cubes: for private person clients (varying over private person relevant dimensions), for organizations and for all clients regardless of their type. We apply the same approach to service dimension. But the support of the application with many cubes is a complex process. The RSM can be used to automate this process.

RSM advantages: The cube splitting will produce cubes without regular sparsity which are denser and decrease an explosion problem.

RSM disadvantages: The produced sub-cubes could still be sparse because of the random sparsity. The RSM based cube partitioning is reasonable only in Multidimensional OLAP environment (in Relational OLAP the cube sparsity does not cause such problems).

4.4. End user facilities

The modification of cube dimensions is a process that leads to series of discomforts not only in relation to the system maintenance, but also for analytical queries and BI end users. One of the consequences of dimensions changes during the life cycle of OLAP application is the effect of the non-relevant nodes. In fact, the BI user is interested in empty cells related to the random sparsity, while the regular sparsity is expected. But when the business scope is a large one it is possible the end user to be not aware of the existing dependences provoking emptiness and he/she to be interested why some cells or regions are empty. In connection with such inconveniences, the knowledge of which part of the cube sparsity is random and which – regular, can be useful for business analysts in various ways:

4.4.1. Automatic selection of relevant dimension elements

The regular sparsity map enables the feature for automatic restriction of user choice of dimension filters or parameters. When a business analyst selects some dimension values, the values of the other dimensions can be restricted to the set of meaningful tuples.

Let us imagine that at a certain moment in time t_k all the supermarkets in our company stop offering product p_j . We have a new rule: “When $time > t_k$ then product $\neq p_j$ ”. In the process of typical business intelligence slice and dice operation, the end user of the system can fix “Time” dimension to t_{k+1} . In fact, he or she is interested in a specific part of the entire cube. The application reads the rules of the map, preprocesses the user request and returns a reduced sub-cube without the p_j layer.

Example

Let the service dimension have the following elements: Travel insurance, Pet insurance, Life insurance, Car insurance.

If we have a business rule “After January 2008, we do not offer travel insurance” and the user fixes the “Time” dimension to February 2008 the available choices of “Service” will be reduced to the available services over the month: Pet insurance, Life insurance, Car insurance.

RSM advantages: Such functionality is absent in most OLAP tools. Wherever it is present there are two disadvantages in comparison with the realization offered by RSM: (a) the process scans the entire cube during every dimension selection query, which can take much time; (b) all elements of the dimension about which there is no data are filtered, and not only the elements in which the user is not interested. The RSM object affords an opportunity for the automatic selection to be implemented (a) without significant query delay and (b) the dimension elements with business meaning to be preserved even when they point empty cells.

4.4.2. Sparsity explanation module

Usually the OLAP application has a feature hiding empty rows/columns in analytical reports or populating them with zeros. However, users often have a different attitude to empty domains induced by regular and random sparsity. On the basis of the regular sparsity map a module that explains the reason (business constraint) why a specific cell is empty can be built. Hiding empty rows caused only

by the regular sparsity and populating them with a “N/A” value also can be implemented. Whereas the empty cells caused by the random sparsity could be populated with a zero.

The replacement of empty cells with zero is related to the question of how you deal with missing data. For example, when a country has no data on wheat production and you're averaging wheat production across a selection of countries, how do you handle this individual country? Do you count it in the total used for averaging, or not? The RSM helps the end users to control a missing values processing in their averages – they can chose to ignore the cells that are part of the RSM or to count them.

Here are our expectations of an explanation module: Imagine that in a typical OLAP report presented in the figure 1 (for April 2008) the user selects an empty cell identified by a service “Pet Insurance” and a region “Varna”. The user is curious why this cell is empty so he calls the explanation module. The module finds out that this cell is in scope of the RSM rule 10 (if service in ...and region in ... then...) and reports: “Pet insurance is not offered in Varna according to rule ...”. Then the user selects an empty cell identified by service “Guarantees”, region “Varna” and asks the same question. The module find out that this cell is not in scope of the RSM at all and reports: “There are no new guarantees in Varna for April 2008”. Also it will be convenient for the module to offer coloring of the cells which are empty due to the quoted rule and/or the cells which are empty due to random sparsity.

RSM advantages: The OLAP tools do not have such functionality. The RSM object is an obligatory factor for its implementation. We also could not found any literature that discussed similar features.

4.4.3. Reports decomposition

The idea of cube partitioning can be used over the user reports. Intelligent splitting of reports with many empty cells to sub-reports without missing values will facilitate the process of business analysis (Figure 1). The compact sub-reports give a better idea of the business dependences.

	Ruse	Sofia	Pleven	Varna	Plovdiv	Burgas
Deposits		6 391		8 038		6 095
Life Insurance	10 298	7 187	9513		8 206	
Guarantees		8 209				6 489
Corporate accounts		5 032		6 056		2 642
Travel Insurance	11 044	6 753	3208		7 625	
Project finance		6 087		6 734		11 003
Pet Insurance	5 578	7 636	9148		11 086	
Car Insurance	5 122	6 355	6803		7 314	

	Varna	Burgas	Sofia
Deposits	8 038	6 095	6 391
Guarantees		6 489	8 209
Project finance	6 734	11 003	6 087
Corporate accounts	6 056	2 642	5 032

	Sofia	Pleven	Plovdiv	Ruse
Travel Insurance	6 753	3 208	7 625	11 044
Pet Insurance	7 636	9 148	11 086	5 578
Life Insurance	7 187	9 513	8 206	10 298
Car Insurance	6 355	6 803	7 314	5 122

Fig. 1. Report decomposition regarding regular sparsity map information

RSM advantages: The important part of the decomposition is the empty cells caused by random sparsity to be preserved. Since there is no an object that preserved such kind of information, the OLAP tools do not have such functionality. For

example, an empty cell for the Guarantees in sub-report 1 could be directly interpreted as “no new guarantees in Varna”, while before the decomposition one have to call explanation module for this purpose. For large reports the explanation module is not very convenient – when the cells are scattered it is impossible to be selected a whole range of empty cells. In addition, after the report decomposition the relation between regions and offered services is obvious.

5. Conclusion and further work

The sparsity of OLAP cubes is a phenomenon in multidimensional data that every designer and database administrator must consider. Sparse data causes the data explosion problem in the precomputation process and decreases the performance of OLAP. The current methods for overcoming of data explosion work mainly on physical level and do not take account of the nature of sparsity. In the present paper we tried to look at sparsity from another point of view – is there some useful information that sparsity can give us? With the help of the newly introduced regular sparsity map we divide the empty cells in a given cube in two categories – empty cells caused by the existence of a real-life business constraint and empty cells caused by a random factor. Then we investigate the applicability of the defined object in three main directions – data quality, performance issues and human interface facilities. Almost all applications of the regular sparsity map object require an algorithm that returns an intersection/union/difference between a regular sparsity map and a rectangular domain. An immediate solution is to store and process the set of tuples covered by the map. Unfortunately, in real-life cases the number of empty cells in a map often exceeds 10^{13} . We think about a method of map preservation that allows implementation of set operation between map and rectangular domains and requires less storage and computational resources. We are in the process of algorithm implementation and look forward to working on further investigation of the regular sparsity map usages.

Acknowledgements: This paper is supported by the Sofia University SRF under Contract 163/2010.

References

1. Casali, A., Cicchetti, R., and Lakhali, L., Cube lattices: a framework for multidimensional data mining, Proceedings of the 3rd SIAM International Conference on Data Mining, San Francisco, California, USA, May 2003 , pp.304-308.
2. Cognos ULC, Improved Performance by Flattening Dimensions, 2008, <http://www.ibm.com/developerworks/data/library/cognos/page174.html>
3. Gyssens, M. and Lakshmanan, L.V.S., A foundation for multi-dimensional databases, Proceedings of 23rd International Conference on Very Large Data Bases, Athens, Greece, August 1997, pp. 106-115.

4. Kang, J.,Yong, H., and Masunaga, Y., Classification of Sparsity Patterns and Performance Evaluation in OLAP System. IEIC Technical Report, ISSN:0913-5685,vol.102,No.209(De2002 61-84), pp.61- 66. Japan, 2002.
5. Michalarias, I., Multidimensional Data Management in Mobile Environments, Ph.D. dissertation, Freie Universität Berlin, Germany, December 2007, pp.11.
6. Naydenova, I., Kaloyanova,K., An Approach Of Non-Additive Measures Compression In Molap Enviroment, Proceedings of the IADIS Multi Conference on Computer Science and Information Systems Lisbon, Portugal, July 2007, pp 394-399.
7. Naydenova, I., Kaloyanova, K., Some Extensions to the Multidimensional Data Mode, Proceedings of the IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing, Sofia, Bulgaria, October 2006, pp. 63 - 68.
8. Nelson, Shayne, The missing data problem, Database and Network Journal, USA, 2005
9. Nguyen, T. B., Tjoa, A. M., and Wagner, R. R., An Object Oriented Multidimensional Data Model for OLAP, Proceedings of the First International Conference on Web-Age Information Management, Shanghai, China, June 2000, pp. 69 – 82.
10. Niemi, T., Nummenmaa, J., and Thanisch P. Normalising OLAP cubes for controlling sparsity. Data and Knowledge Engineering, 46(1):317-343, September 2003
11. Oracle Corporation, Oracle9i OLAP Developer's Guide to the OLAP DML Release 2 (9.2), Part Number A95298-01, March 2002, pp. 3-18.
12. Pedersen, D., Riis, K., and Pedersen, T.B., A powerful and SQL-compatible data model and query language for OLAP, Proceedings of the 13th Australasian Database Conference, Melbourne, Australia, January 2002, pp.121 - 130.
13. Pendse, N.,Database explosion, <http://www.olapreport.com/DatabaseExplosion.htm>, Business Application Research Center, Feb 2005.
14. Pendse, N., The Problems with OLAP, DM Review Magazine, March 2007.
15. Potgieter, J. OLAP Data Scalability, DM Review Magazine, October 2003. Available at: <http://www.dmreview.com/dmdirect/20031031/7636-1.html>
16. Rahm, E.,Do, H.H.,Data Cleaning: Problems and Current Approaches, IEEE Techn. Bulletin on Data Engineering, Dec. 2000, p.11, (2000)
17. Singh, A.,Handling inter-dimensional members dependency and reducing cube sparsity using reference dimensions in Analysis Services 2005 SP2, May 2008, <http://asmdx.blogspot.com/2008/05/handling-inter-dimensional-members.html>

SAS Data Warehouse and its Usage in Government Public Sector

Snezana Savoska¹, Violeta Manevska², Silvana Kolevska³,

¹ Faculty of Administration and Management of Information systems, University „St.Kliment Ohridski“, Partizanska bb, 7000 Bitola, R.of Macedonia , www.famis.edu.mk

Abstract: Data Warehouses are one of the most complex IT issues whose implementation demands much organizational preparation. In fact, its successful implementation demands specified levels of organizational culture and management’s full support. Nevertheless, when data warehouses are owned by the government, a very professional approach in the creation of data warehouses, as well as usage of procedures for creating specifically prepared information for dissemination in the public sector, are imperative. Our research on the application of data warehouses in the Republic of Macedonia was focused on the biggest business companies, some governmental institutions and higher education institutions. We found out that there is an increasing usage of data warehouses especially in the business sector. However, the most complex data warehouse in Macedonia is the one with the State Office of Statistics (SOS). This is the backbone of the Macedonian public administration which aims at gathering, analyzing and dissemination of relevant information for entities which are living and working in the Republic of Macedonia. This implies that it is a very important state institution accountable for collecting and storing official information with a very serious approach towards data warehousing information for statistical data analysis. The serious role which has been attributed to this state institution requires a professional software tool and that is why SAS Data integration services tool has been selected.

Keywords: Data Warehouse, State Office of Statistics, Data visualization.

1 Introduction

Data Warehouses are one of the most complex IT issues which demand much organizational preparation for its implementation. First of all, the organizational culture must be minimal on the defined level and, also, there must be management’s full support for successful implementation. When the government is the owner of a data warehouse, there must be a very professional approach in the data warehouse creation and also the usage of procedures for the creation of specifically prepared information for the public sector. Our research on the usage of data warehouses in the Republic of Macedonia was focused on the biggest business companies, as well as some governmental and higher education institutions. We found DW solutions from



the biggest IT global company as Oracle, SAS, SAP, and Microsoft. There is an increasing application of data warehouses especially in the business sector, where we have identified usage of OLAP multidimensional databases and Analysis services as pivot tables and as end-user's reporting tools.

2 The usage of DW in the public administration

However, the most complex data warehouse in Macedonia was located in the State Office of Statistics (SOS). SOS is the cell of the Macedonian public administration which aims at gathering, analyzing and dissemination of all relevant information for entities which are living and working in the Republic of Macedonia. This fact implies that this is the most important state institution accountable for collecting and storing of official information. It has an extremely serious approach in collecting, cleansing, analyzing and storing information for statistical data analysis. This serious role attributed to this state institution requires a professional software tool for DW.

Data which are included in SOS DW usually are taken from many administrative sources; areas which gain data on the basis of statistical research carried out by SOS, as well as data from the Statistical Register in SOS.

3 The data sources for State Office of Statistic of DW

The administrative sources comprise the Central Register of the Republic of Macedonia (subjects' final finance reports); Public Revenue Government Service (taxes information database), Ministry of Finance (public revenue and expenditure – treasure account), National Bank of Macedonia (monetary statistics), Health Insurance Fund (health insurance contributions) as well as Pension and Disability Insurance Fund (pension and disability insurance contributions).

The statistical research comprises the areas of foreign trade, internal trade, industry, construction and hospitality, agriculture, household consumption surveys, price statistics and labor force survey.

As a source of statistical registers SOS uses the Register of business entities, which is used in the stages of data preparation for loading data in the DW.

The entire operation of the DW is carried out in the SAS DW environment. The source of data for DW are previously taken from defined data sources and prepared with Microsoft Visual Studio.NET (Visual basic and ACCESS) or some other procedures for data preparation. The tool used for importing data from the prepared data sources (saved in the auxiliary lactation) is SAS Data integration studio (Figure 1). It is a very powerful tool which provides working with many data repositories located in data servers or external sources. Procedures made for this purpose are used for incorrect and missing data; their transformation; conversion of characters written with the Macedonian language support in ACSII codes and are prepared in format apposite for its first loading in the DW (first run). These procedures are created with SAS BPM [3, 4] tools with manual code corrections and they aren't completely

automated (because it is impossible to make them automated to a high degree). They are presented with modeled software business processes (Figure 2).

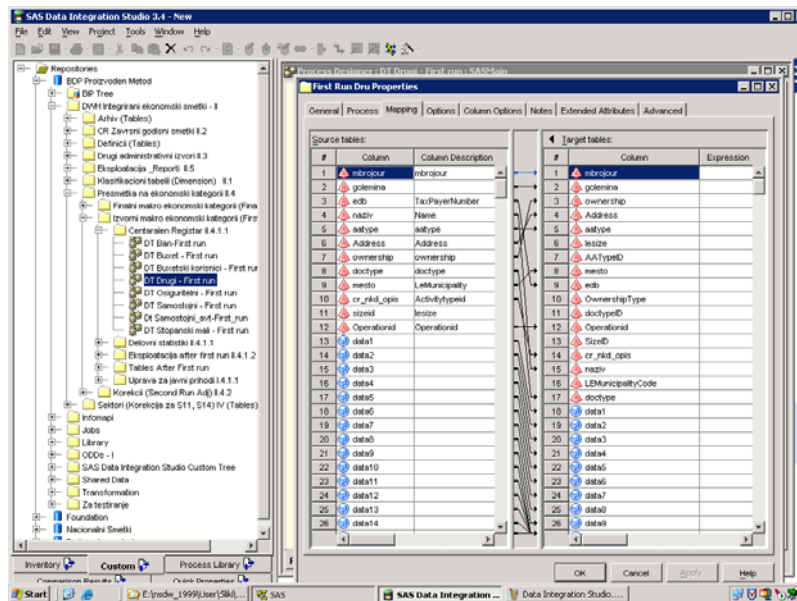


Fig. 1. SAS Data Integration services screenshot

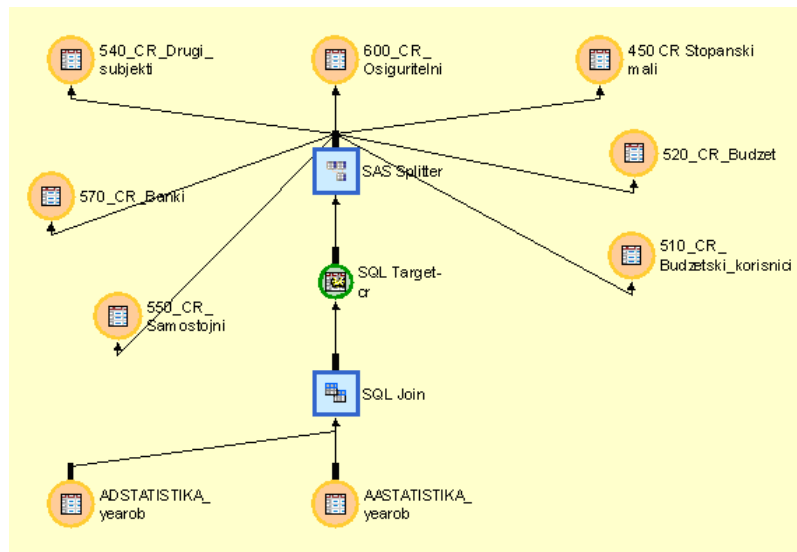


Fig. 2. Modeled software business processes which are objects with embedded program code

4 Metadata of SAS Data Warehouse

The metadata can be different depending on data types – source data, its cleansing and transformation, intermediate data and target data, its aggregations and the manner of preparation for the final run processes. The data can be described in different manners. In this case, all metadata descriptions, source data transformations, business processes preparation for the different phases are located in the Excel file named Categories.xls. In this complex file, there are business logic procedures, statistical analysis demands and algorithms for demanded aggregated tables creation and reports located (Figure 3). Moreover, this file contains intermediate calculations' results. Some of the products of these calculations are located in the temporary file directories whereas the other ones are located in annual data processing procedures and annual data warehousing locations.

Label	Karakter	AOP	ESA	tip	firstrun	secondrun	finalrun
Bruto vrednost na proizvodstvo	sum(Bruto_k.+trgovska_marza)		P1				Bruto
Pazarno proizvodstvo	Bruto		P11				
Mejufazna potrošnja	sum(megufazna_k.-korekcija_plati)		P2				Megufazna
Bruto dodadena vrednost	sum(bruto.-megufazna)		B1g				Bruto_dodadena
Plati i naemmini	sum(data344.+data740.+data694.+data896.+data700.+data8702.+data3706.+korekcija_plati.+korekcija_bruto)		D11				Plati
Pridonesi od plati	data738		D12				Pridonesi
Sredstva na vrabotenite	sum(sredstva_i.+korekcija_plati.+korekcija_bruto)		D1				Sredstva
Amortizacija	data348	269	K1				Amortizacija
Neto dodadena vrednost	sum(Bruto_dodadena.-amortizacija)		B1n				Neto_dodadena
Bruto deloven višok	sum(bruto_dodadena.-sredstva)		B2g				Bruto_visok
Neto deloven višok	sum(Neto_dodadena.-sredstva)		B2n				Neto_visok
Broj na vraboteni	sum(vraboteni_i.+vraboteni_ars)						vraboteni
Po-etni vkupni zalhi	data89	23			p_vk_zalhi		
Krajni vkupni zalhi	data92	23			k_vk_zalhi		
Po-etni zalhi na proizvodstvo i gotovi proizvodi	data642	652			p_proizvodstvo_proizvodi		
Krajni zalhi na proizvodstvo i gotovi proizvodi	data644	653			k_proizvodstvo_proizvodi		
Promena na zalhite na nedovr.proizvodstvo i gotovi proizvodi smetkovodstvena vrednost	sum(data644.-data642)				proizvodstvo_proizvodi		
Po-etni zalhi na materijali, trgovski stoki i avansi	sum(data89.-data642)				p_saldo_materijali_stoki		
Krajni zalhi na materijali, trgovski stoki i avansi	sum(data92.-data644)				k_saldo_materijali_stoki		
Promena na vkupnite zalhi smetkovodstvena vrednost	sum(data92.-data89)				zalhi_bez_ceniP52		

Fig. 3. Excel file Categories.xls which is the core file for defining metadata and business processes in the DW

5 The stage of Data Transformation and DW updating

In the stage of DW updating with the business entities' final financial reports, the procedure includes merging data for each entity with fact and dimensional tables of DW. The data from the database of the Central Registry are obtained in the ACCESS

database tables which are saved in the source data folders and then they undergo a procedure for data transformation in the format comprehensive for SAS DW. These dimensional tables contain the territorial classification which is valid for the Republic of Macedonia; the institutional sector to which the data belong; Classification of National Activities (CNA) etc. This data are the base for new calculations of the variables, called categories for GDP calculation (gross value, intermediate consumption, value added, number of employees, assets employed, depreciation etc). The schematic display of the processing of source data until reaching final calculated data is shown on Figure 4, whereas the grouping of aggregated data according to sector affiliation is presented on Figure 5.

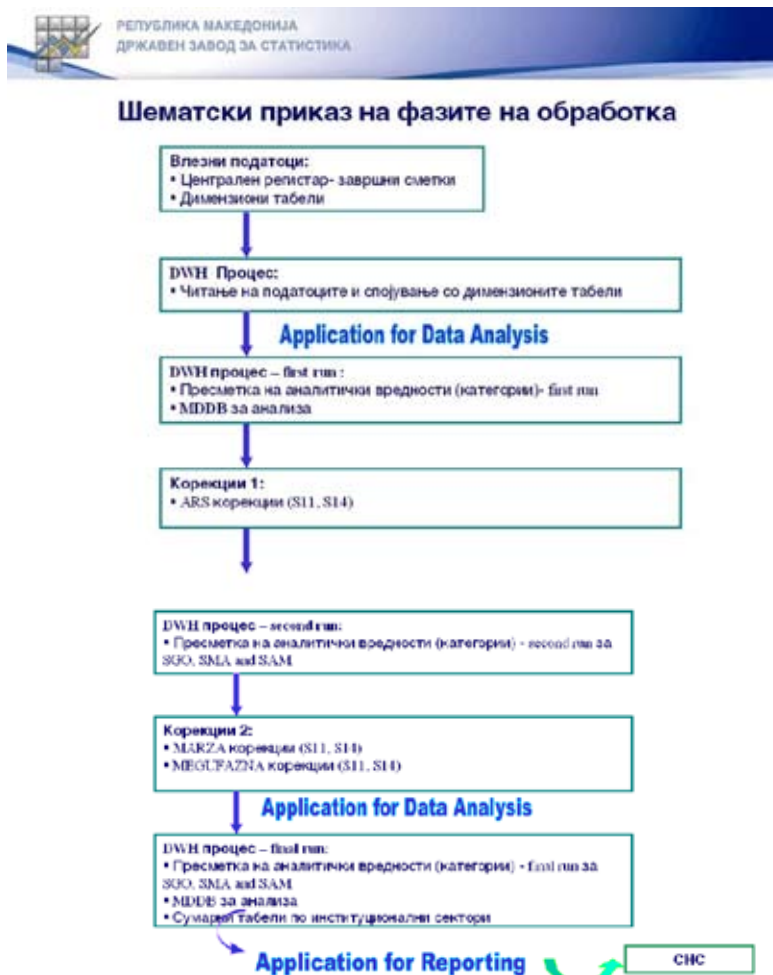


Fig. 4. Phases in the process of data preparation for DW loading stage

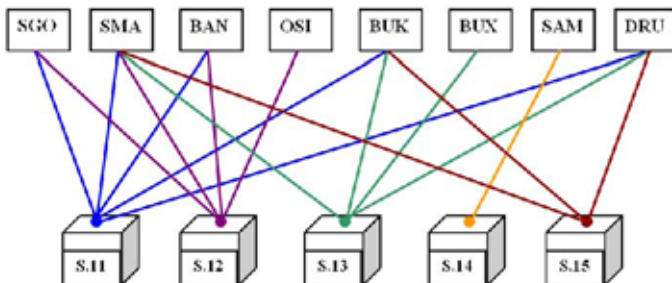


Fig. 5. Aggregated tables according to state institutional sector

SAS DW owns many tools which have different functions. The SAS ETL tool possesses many other parts whose licenses are sold separately. The SAS products which are used in SOS are: SAS Base; SAS Data integration studio; SAS Adjustment analysis; the tools Corrections and Reporting (Figure 6).



Fig. 6. Screenshot of Adjustment Analysis tools, Corrections – Data Analysis

Dissemination of data prepared in DW is another aspect [5] of the end user's data usage. For this purpose, the most suitable tool is the SAS Reports tool (Figure 7). These tools provide selection of many dimensions, time periods and lists of offered reports.

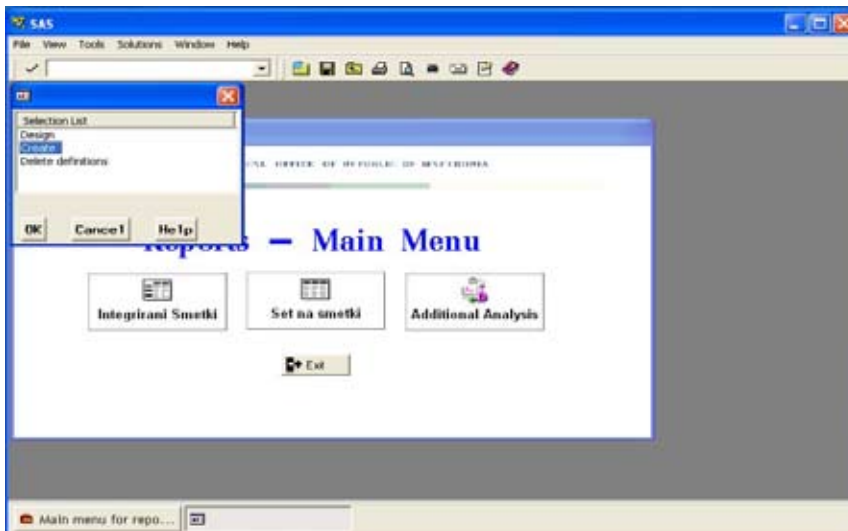


Fig. 7. Screenshot of SAS Reports main menu

6 Publication of data from SSO Data Warehouse

However, the publication of the prepared data in the DW is not a DW administrator's task [1]. Dissemination of public data is entrusted to the SOS sector for public relation whose mission is to provide public access to SOS data via SOS Web site (<http://www.stat.gov.mk>). The web site is a great combination of information intended for public usage. The data from DW can be seen with the choice "Databases" where we can find statistical analysis of data of interest combined in tabular format and appropriate visual format. The user access to the web viz browser with a modern HCI provides selection of desired data, sectors, geographic location and format of report. When the desired selection is filled, the data are extracted from DW flat and dimensional tables (mostly from aggregated tables previously prepared in the DW depending on suitable procedures). There is a tool offered for specific data visualization format PC-AXIS for Windows, which can be downloaded from the site and which provides a special format compatible with EU statistics standard tools which can give maps, graphs, diagrams and tables [2]. Figure 8 presents an instance of a screenshot of some statistical population data (visual region representation, statistical data for specific characteristic and thematic scope of categories which are gained from selections www.stat.gov.mk).

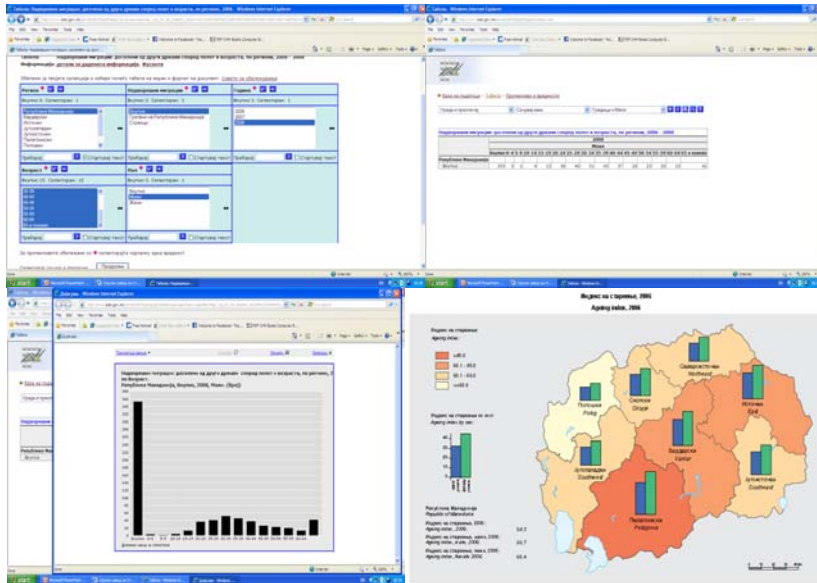


Fig. 8. Example of SOS web site – step by step

7 Conclusion

When talking about most complex data warehouse in Macedonia, we think about SAS DW. It is located in the State Office of Statistics (SOS) and is a cell of the Macedonian public administration. Its purpose is gathering, analyzing and dissemination of all relevant information for entities which are living and working in the Republic of Macedonia. This serious role attributed to this state institution requires a professional software tool for DW and also professional tools for data dissemination via web site.

References

1. D.Keim, H.P.Kriegel, M.Ankerst, Recursive Patterns: A Technique for Visualisation Very Large Amounts of Data, Proceedings of IEEE Visualization '95, 1995, pp. 279-286
2. J.Becker, D.B`urkle, R.T.Happe, T.Preußer, M.Rumpf, M.Spielberg, R.Strzodka, Aspects on Data Analysis and Visualization for Complex Dynamical Systems, 2000
3. www.sas.com, SAS® OLAP Server, www.datawarehouse.com
4. www.datawarehouse.com, SAS Extends Support for Open Metadata With Meta Integration Technology, Inc. , 7.3.2006
5. J.Yang, et al. , "Value and Relation Display for Interactive Exploration of High Dimensional Datasets", Proceedings of InfoVis '04, Austin, TX, Oct. 2004, pp. 73-80
6. www.stat.gov.mk, 2010, Web site of State Office of Statistics

Business rules extraction and management- problems and solutions

Evgeny Krustev,
5, James Bouchier blvd., 1164 Sofia, Bulgaria
eck@fmi.uni-sofia.bg

Abstract. There is a huge number of organizations employing software with a legacy problem that is not allowing an easy update of the business rules according to the changing business environment since the business rules are hard encoded in the source code. This paper considers approach for extraction and management of business rules from such systems on the basis of Semantics of Business Vocabulary and Business Rules standard of OMG. A broad range of techniques are considered both in business rule extraction and business rules management, their limitations are discussed and new trends in solving that problem are outlined.

Keywords: business rules, legacy system, semantics, SBVR standard, business rules extraction, program slice, business rules management.

1 Introduction

This paper considers the problem of business rules (BR) extraction from legacy systems, which has attracted a lot of academic interest once it got necessary such systems to be updated or even “redeveloped” in order to satisfy the modern business objectives of the owning organization. Usually information technology that is about five years in production has such a legacy problem due to the innovation cycle in software technology. The term legacy software generally refers to software written C, COBOL, FORTRAN around the 1970’s, although currently software, written in the late 1990’s including early Java programs, is also considered as legacy software [1]. Additionally, organizations tend to cut investments in new technology in times of a global economical and financial crisis and prefer to find cost efficient solutions on the basis of the technology they already use. One such solution appears to be the extraction of the existing BR from legacy source code and reengineering that software[2-8] to meet current business goals, while recently the transformation of existing legacy software into reusable components[1, 9-10] or extracting BR through process mining [11] are being preferred.

A common feature of the design pattern employed to build legacy software is that the BR are hard encoded in such systems due to the software technology used at the time. Therefore, it is more appropriate classifying software as “*legacy*” not that much on the basis of the time it has been put in production rather on the “*legacy*” type of software architecture used in such systems.



Until the adoption in 2005 of the Semantics of Business Vocabulary and Business Rules (SBVR) as a standard by OMG [12], there has been no commonly accepted understanding about the semantics of business rules. Moreover, usually in the course of time the discrepancies between documentation and the actual state of the legacy software it relates to are making it impossible to correctly identify the pieces of code that implement one or another business rule in order to validate or update it. In fact, in most cases the BR have not been documented at all as at the time there has been no standard software engineering approach for this purpose. Thus the method of program slicing [13] has been introduced in an effort to facilitate the extraction of BR.

In the last 10- 15 years a lot of research has been done in the area of BR extraction from legacy software, while currently there are only few tools (see the Appendix A of [5]) available for commercial applications and research. The application of such tools in the case of large and complex legacy systems still remains a difficult task. The objective of this paper is to outline some of the problems in this respect and identify possible solutions that would allow easier management of BR after the reengineering or system transformation process of legacy software is complete.

2 Business rules

The process of extracting BR requires first to define what is understood by a business rule. Several different definitions for a business rule have been used until the SBVR standard has been by OMG [12]. This standard not just defines a business rule, but relates it to business vocabulary. Thus all business rule expressions are based entirely on business vocabularies. Hence, business rule are independent of all processes and events. According to the SBVR standard a business rule is a “*rule that is under business jurisdiction*” and it satisfies all three of the following criteria [14-16]:

- a) Represent a guide for business conduct or business action including elements of obligation and necessity. This describes a business rule as “*actionable*” and in simple words a business rule must clearly define what do or not to do in a particular business situation.
- b) The rule statement must be about business and depend only on the business vocabulary. Hence, BR are implementation independent. According to this criterion rules that depend on a business process, a system that support the business or a platform that implements such a system are not regarded as BR. Thus, business requirements, software requirements, integrity constraints etc [17- 19] are not considered as BR.
- c) The rule must be expressed in an English declarative statement expressing the view of business. It is common to wrongly express a business rule in terms of a procedure making use of the common *if- then-else* IT style [20- 21]. Adhering to a set of guidelines named *RuleSpeak* [22] for expressing BR helps in achieving greater clarity and consistence in communicating rules.

The above allows us make the conclusion that not all BR may be found in source code and that we cannot trust the source code for a correct implementation of the BR. The BR extracted from the legacy systems can only hint about the actual BR implemented in that systems. Even when a business rule has been extracted from the

legacy system and verified to be a really a business rule it has to be stated as a declarative statement from business point of view so that it could be correctly managed later on.

To complete this short review of BR we have to mention that BR fall in two fundamental categories [12, 14] from the business perspective:

- Structural rule (also called *definitional*). Rules in this category usually imply a sense of necessity or impossibility. In terms of the *RuleSpeak* guidelines the rules from this category are recognized by using the keywords “*always*” or “*never*” as part of their statement. This way a structural rule always classifies or computes something.
- Operative rule (also called *behavioral*). Rules in this category usually imply a sense of obligation or prohibition. In terms of the *RuleSpeak* guidelines the rules from this category are recognized by using the keywords “*must*” or “*only*” as part of their statement. All BR that can be directly violated by people involved in the business environment are referred to as operative rules and sometimes such rules are also called “*people rules*”. These are rules that allow the business operate in optimal accordance with its goals by referring to certain cases of business operation that might affect in a negative way the business objectives.

The SBVR standard implies BR are built on facts that are expressed by means of terms, where both the terms and the facts are part of the business vocabulary. Especially closely related to the business vocabulary are the operative rules. A summary of research on papers that consider most often legacy systems written in C and COBOL[7,16] mainly use structural rules as an example for the effectiveness of the proposed algorithm or approach for BR extraction, while they miss to identify or ignore the operative rules. It happens most of the time, when the business vocabulary is not well presented, as well as, because such rules deal with reasoning and process models that are difficult to extract automatically from source code.

The above identified problems necessitate a formal approach to extraction of BR.

3 Extraction of business rules

Business rule extraction is a complicated procedure that is comparable to an archeological dig [2]. Unlocking the secrets of BR hidden in legacy systems requires the investigation mainly of three resources- the legacy source code, the available documentation and people. Since the information collected from all these sources may be incomplete usually all formal approaches toward rule extraction require an iterative procedure including the interaction with knowledgeable business analysts. [3, 5, 9, 16, 23].

Similarly to archeological work the BR extraction starts by a preliminary analysis of the existing sources- filtering out and tagging important information, removal of irrelevant code, determining data structures with impact on BR etc. It is a necessary step that allows the actual work on BR extraction to execute more efficiently. At this stage the Business vocabulary is created and loaded with primary business terms and business facts, as well as, a candidate set of business rules is established. Once the

preliminary analysis is complete an iterative procedure over the following steps usually takes place:

- a) Update the Business vocabulary by adding to it business terms and business facts discovered by investigating the main resources (source code, documentation and the people involved in the system operation). Clues about terms and facts are usually found in program names of variables and data types, labels in the user interface and reports. Existing tools for gathering such information in the so called “boundary elements” of the application allow to automate this process, however, the role and active assistance of business analysts in completing this task cannot be replaced.
- b) Update the set of extracted BR by investigating code fragments affected by terms or facts included in the Business vocabulary. For this purpose the expertise of a business analyst and appropriate tools for an in-depth analysis of the data flow and control flow of the program are needed.

Most often the investigation of the source code, that hints for elements of the Business vocabulary and for potential BR, involves the computation of a program slice[13] comprising an executable subset of the program code that affects just the value of a set of variables at a selected statement of the program referred to as slicing criterion. Slicing techniques have a wide scope of application and it has sparked a lot of research interest, for instance, around the Wisconsin Program-Slicing Project [24] or the Indus project [25]. There are slicing tools [5] written for C, COBOL, FORTRAN, OBERON- 2 [26] and recently such tools have emerged also for Java, where the suggested solutions [25, 27] employ efficiently the object- oriented features of the Java language including multithread applications. In the context of rule discovery such tools allow automated tracing of logic paths subject to various selection criteria. However, the implementation of these techniques on large legacy systems in an effort to discover the BR faces certain difficulties. For instance, a large system naturally yields a large program slice and provides little or no information about the BR. Of course, there are performance issues, as well. Heuristic rules and enhancement of the slicing criterion are proposed [6,8] to improve somehow the performance the program slicing procedure. Unfortunately, in the literature there exist quite general qualitative performance comparisons of one program slicing algorithm with another one as part of a BR extraction case.

Program slicing at the source code level analysis has certain inherent limitations due to treatment of behaviors unspecified by the semantics, for instance, pointer arithmetic and indirect calls or uninitialized variables in C/C++. Such unspecified behaviors complicate a lot BR extraction from large systems. Therefore, recently new dimensions for rule extraction have been discovered- executable code analysis [29], mining a repository of process designs [11].

Once the BR are extracted they have to be documented. It is the last step of the BR extraction and it is most often omitted in papers dealing with BR extraction. In the best case such papers normally present the extracted rules in an IT- friendly style. The usage of templates for documenting the extracted BR [28] enforces consistency in expressing the BR subject to the adopted SBVR standard.

In summary, the SBVR standard provides the basis for a uniform procedure for BR extraction, while any particular implementation of this procedure requires expert

knowledge in interpreting the output from multiple tools used to discover clues about Business vocabulary terms and facts, as well as, identify BR candidates. A trend in such tool development shows a shift toward integration of business extraction with modern business rule management technologies allowing service oriented architecture of the reengineered legacy product.

4 Business rule management

Once the BR are being extracted from a legacy system then the problem of their management gets important. Unlike the times when such systems have been coded now there is clear understanding about the need to manage BR separately of their implementation. Indeed, business rules represent decisions about how to run business and decisions change frequently even more frequently than in the past. Missing to “*separate concerns*” when managing the extracted business rules from a legacy system would result in another software system of legacy type.

To avoid the mistakes of the past successful BR management employs Business Rules Management technology (BRM). Currently there are many BRM systems tailored for Java [30] and .NET [31]. A common feature for BRM systems is the availability of a Rules engine to store BR similarly to data, modify and maintain BR employing a declarative programming language. Among the rest it provides the following advantages:

- BR are distinctly separated from their implementation in procedural code. It eliminates the need to extract BR from procedural code
- Store BR in a database and update them while the application is running
- Employ rule templates where rules are conditionally generated on the basis of values of a predefined set of data
- Less time is needed to add new BR or update existing BR , while some systems don’t need to get recompiled in order to execute the updated BR
- Enables easier management and understanding of BR by business analysts through user friendly graphical interface

The benefits of employing the BRM are illustrated in [23] and [32]. Both demonstrate the use of a rules modeling tool to store and manage rules. In particular, [23] provides multiple business rule program code implementations for a given business rule, while [32] gives an example of using rule templates in the telecommunication business.

In summary, modern BRM system enable significant benefits in BR management. It allows efficient management of the BR once extracted from legacy systems.

5 Conclusion

This paper considers the models and the techniques used to solve the problem of extracting BR from legacy system and management of the extracted rules. Software systems with hard encoded BR are here referred to as legacy systems. This design

pattern proves to be a heavy legacy problem complicating the extraction of business rules in the general case when the documentation and expert knowledge about the business rules is incomplete or missing.

The great practical interest in that subject has contributed to the active research in this area. As strange as it is the driving force for that interest is the basic reason for the legacy problem itself. From a business point of view the business rules should be adapted to the changing and highly competitive business environment. A cost efficient solution for legacy systems is to extract the business rules on the basis of investigating the legacy system code and reengineer that system employing modern technologies for BR management by separation of concerns (BRM system). Another alternative is to design the business processes and employ a BRM system for both extraction and management of the BR.

In this respect the paper focuses on the importance of using the SBVR standard of OMG. It allows presenting a unified standard- based approach to extracting and managing business rules in this paper. A broad range of techniques have been analyzed, their limitations and innovative trends have been identified on this basis.

Acknowledgments. This paper is supported by the Sofia University SRF under Contract 163/2010.

References

1. Sneed, H., : Integrating legacy Software into a Service oriented Architecture, Proceedings of the Conference on Software Maintenance and Reengineering (CSMR'06), IEEE (2006)
2. Ulrich, W.,: Legacy Business Rule Capture: Last Piece of the Transformation Puzzle (2009)
http://www.systemtransformation.com/IT_Arch_Transformation_Articles/arch_legacybrc.htm
3. Ulrich, W.,: Extracting Business Rules from Existing Systems (2005),
http://www.systemtransformation.com/arch_extracting_business_rules.htm
4. Ulrich, W.,: Knowledge Mining: Business Rule Extraction & Reuse,
http://www.systemtransformation.com/IT_Arch_Transformation_Articles/arch_knowledg.htm
5. Ulrich, W.,: Legacy Systems: Transformation Strategies Prentice Hall, (2002),
http://www.systemtransformation.com/legacy_book.htm
6. Wang, C., et al., : Extracting Prime Business Rules from large legacy system, Proc. of the Int. Conference on Computer Science and Software Engineering. IEEE (2008)
7. Harry M. Sneed, H., Erdos, K., : Extracting Business Rules from Source Code, Proc. of the 4th International Workshop on Program Comprehension (WPC '96) IEEE (1996)
8. Huang,H., et al.,: Business Rule Extraction from Legacy Code, Proceedings of the COMPSAC '96 - 20th Computer Software and Applications Conference, IEEE (1996)
9. Chu Chiang, C., :Extracting Business Rules from Legacy Systems into Reusable Components, Proceedings of the IEEE/SMC International Conference on System of Systems Engineering Los Angeles, CA, USA, IEEE (April 2006)
10. Akers, R., Baxter, I., Mehlich, M., Ellis, B., Luecke, K., : Re-engineering C++ Component Models Via Automatic Program Transformation, Twelfth Working Conference on Reverse Engineering, IEEE (2005)

11. Polpinij, J., Ghose, Aditya K., Hoa Khanh Dam, : Business Rules Discovery from Process Design Repositories, Proceedings of the 6th World Congress on Services IEEE (2010)
12. Semantics of Business Vocabulary and Business Rules (SBVR) Version 1.0 Specification, <http://www.omg.org/spec/SBVR/1.0/>
13. Weiser, M. D.,: Program Slicing. IEEE Transactions on Software Engineering, 10, pp. 352- 357 (July 1984)
14. Ross, R.,: What Is a Business Rule?, Business Rules Journal, Vol. 11, No. 3 (Mar. 2010), <http://www.BRCommunity.com/a2010/b525.html>
15. Ross, R.,: Are all Rules Business Rules? Not!, Business Rules Journal, Vol. 8, No. 5 (May 2007), <http://www.BRCommunity.com/a2007/b345.html>
16. Baxter, I., Hendryx, S. A Standards-Based Approach to Extracting Business Rules, OMG's Architecture Driven Modernization Workshop, Semantic Designs (2005) <http://www.semdesigns.com/Company/Publications/ExtractingBusinessRules.pdf>
17. Ross, R.,: Are Integrity Constraints Business Rules? Not!, Business Rules Journal, Vol. 8, No. 3 (March 2007), <http://www.BRCommunity.com/a2007/b335.html>
18. Ross, R.,: Are Software Requirements Rules? Not!, Business Rules Journal, Vol. 8, No. 4 (April 2007), <http://www.BRCommunity.com/a2007/b341.html>
19. Gladys S. W. Lam, : Business Rules vs. Business Requirements, Business Rules Journal, Vol. 7, No. 5 (May 2006), <http://www.BRCommunity.com/a2006/b290.html>
20. Ross, R.,: What's Wrong with If-Then Syntax For Expressing Business Rules. One Size Doesn't Fit All, Business Rules Journal, Vol. 8, No. 7 (July 2007)
21. Ross, R.,: The Business Rules Manifesto v. 2 (2003), <http://www.businessrulesgroup.org/brmanifesto.htm>
22. RuleSpeak, <http://www.rulespeak.com/en/>
23. Khusidman, V.,: Business Rules Discovery from Existing Applications, Business Rules Journal, Vol. 8, No. 10 (Oct. 2007), <http://www.BRCommunity.com/a2007/b366.html>
24. The Wisconsin Program-Slicing Project. <http://www.cs.wisc.edu/wpis/html/>
25. Venkatesh Prasad Ranganath, John Hatcliff : Slicing concurrent Java programs using Indus and Kaveri , International Journal on Software Tools for Technology Transfer (STTT) Volume 9, Numbers 5-6, 489-504 (2006)
26. Steindl, C.: Program Slicing for Object-Oriented Programming Languages, Dissertation in Computer Science at the Johannes Kepler University Linz, Austria (1999), <http://www.ssw.uni-linz.ac.at/General/Staff/CS/Research/Publications/Ste99a.html>
27. Bi-Xin Li, Xiao-Cong Fan, Jun Pang, Jian-Jun Zhao, :A model for slicing JAVA programs hierarchically , J. of Computer Science and Technology Vol. 19, No. 6 pp. 848-858 (2004)
28. Reeder, J., :Templates for Capturing Business Rules ,Business Rules Journal, (April 2001)
29. Balakrishnan, G., : WYSINWYX: What You See Is Not What You eXecute. Ph.D. dissertation and Tech. Rep. TR-1603, Comp. Sc. Dept., University of Wisconsin, Madison, WI (2007) http://www.cs.wisc.edu/wpis/html/#kidd_thesis
30. Links to business rule management tools, <http://www.businessrulesgroup.org/linkbrg.shtml#tool>
31. Bukovics, B., : Pro WF: Windows Workflow in .NET 4, Apress (2010)
32. Biying Lin, Yanhui Zhang, : The Management and Validation of Business Rules in Telecom Operator's Settlement and Apportion System, Proc. of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing IEEE (2007)

Modeling of Primary Copy Two-Version Two Phase Locking

Svetlana Vasileva¹

¹ Shumen University „Bishop Konstantin Preslavski“, College – Dobrich, Dobrotica 12, Dobrich, 9302, Bulgaria
svetlanaeli@abv.bg

Abstract. The method of two-phase locking (2PL) is still used in concurrency access of the global transactions management. It has its advantages in distributed database systems (DDBMS), conflict saturated events. The paper views a model of a variant of the protocol for 2PL with primary copies for DDBMS using two versions. As in concurrency control of distributed transactions in DDBMS deadlocks are possible to appear, one of the ways for their preventing is the usage of two-version architecture of the data in distributed database (DDB). The lack of algorithms published as well as results from investigations of such algorithms impose the development of simulation models and their investigation. In detail the paper considers such a model of Two-version two-phase locking (2V2PL) with primary copies. The model is created by means of GPSS World Personal version. The simulations of the algorithm and the results reported are achieved by the system of imitation modeling.

Keywords: distributed transactions, distributed database, concurrency control, two-version two-phase locking, GPSS blocks

1 Introduction

As a result of the data fragmentation and application the throughput of the system increases in the distributed database management systems (DDBMS). One of the basic problems in this type of systems is the concurrency control to shared data of concurrent execution of transactions. In systems saturated with conflict events of the techniques for parallelism management, the most effective method is the one of the locks (according to a great number of authors). Therefore, it is desirable to use the two-phase locking (2PL) method. The problems which appear during its application need to be investigated. One of the effective and cheap investigation methods for different types of systems work is the simulation method.

This is the reason we consider such an approach in our paper to develop modeling algorithms for management of transactions parallelism, which is based upon the protocols for 2PL in distributed databases (DDB). The development is grounded on

¹ This paper supported by Project N ПД-07-987 of Shumen University “Bishop Konstantin Preslavski” whose topic is Software models and systems in pedagogical and scientific studies



the presented and considered in [4] and [6] simulation algorithms of centralized two-version 2PL and distributed two-version 2PL. In the modeling algorithms the data undergo incomplete replication (all the data elements have one and the same number of copies, the number of the copies of a data element is smaller than the number of the sites). The simulation models are developed by means of the classical modeling system - GPSS World.

As in pessimistic protocols deadlocks of transactions are possible to appear, some problems to find and solve the locks come up. One of the ways to avoid the locks is to use multi-version architecture of the data in DDBMS. [2] presents and considers multi-version algorithms for concurrency control in database management systems (DBMS) and in particular: Multi-version 2PL and Two-version 2PL.

2 Version Algorithms for Two-Phase Locking

There are [2] two main protocols for two-phase locking in DBMS, which protect the transactions from deadlocks in their concurrent service.

2.1 Multi-Version Two-Phase Locking (MV2PL)

There are three types of versions of data elements: committed versions – created by these transactions which have already done their work successfully; current version – the last of the completed versions; uncommitted versions – versions created by these transactions which are still working. The MV2PL scheduler follows the things so that there would be no more than 1 incomplete version in each moment of time. All operations which the Lock Manager (LM) processes are divided into two classes: regular and final operations. Final is the last transaction operation before its completion or the operation of fixing itself.

Each separate operation is processed by MV2PL scheduler the way described in [2]:

1. If the operation is not final then: (a) operation $r(x)$ is executed immediately as it is compared to the last of the fixed up to the moment versions of x ; (b) $w(x)$ operation is executed only after the fixing of the transaction which has recorded the last version of x .
2. If the operation is final for the T_i transaction then it is delayed up to the moment any of the following transactions gets completed: (a) all T_j transactions that have read the current version which has to replace the version recorded by T_i ; (b) all T_j transactions which have recorded versions read by T_i (where the last operation is the operation for fixing) x .

Serving in this protocol shortens the time for the stay of the transactions in the system significantly and decreases the possibility for transactions deadlocks considerably.

But during the realization of the protocol, enormous problems arise. They are related to the fact that the number of the versions of one and the same element which can exist concurrently in DB is not followed. The storage of the versions could possibly need a lot of memory as the number of the versions is not preliminary

known, it is difficult and complex to realize an effective structure of their storage for which not to have that big expenses. Such a system is quite complex to realize.

2.2 Two-Version Two-Phase Locking (2V2PL)

The Two-Version Two-Phase Locking protocol is suggested for the first time by Bayer in [1]. In the protocol of 2V2PL there are only two versions of the element: 1 current version of the element and no more than one incomplete version.

Three types of locks are used as each lock is released at the end of the transaction: rl - (read lock) – it is assigned on the current version of the data element immediately before its carrying out; wl – (write lock) – it is located before the new (incomplete) version of the element is created; cl - (commit lock) – it is located before the execution of the last operation of the transaction (usually before the commit operation) on each data element which it has recorded.

The advantages of the protocol: the limit of the number of old versions to one reduces the volume of DB; It is not necessary to have a special storage structure of the versions; a simple enough protocol to realize.

Disadvantage: such organization of the versions is beneficial for the transactions executing the operation of reading.

Problem: The issue about transactions executing the writing operation is not effectively solved.

The development and realization of the multi-version algorithms (MultiVersion Concurrency Control - MVCC) solves many of the problems in mono-version algorithms for concurrency access management but not all of the developed version algorithms are put actively in practice which is due to the problems referring to the creating of version DBMS, and namely:

- Necessity of building physical organization of the data which provides effective management of the versions – report: of the volume expansion of DB in versions addition; of the possible limits on the number of versions, but in practice this limitation often lacks – the system simply deletes the old versions;

- Necessity of reporting the versions in other components of DBMS as well.

In practice the most often use is that of the Protocols Read-Only MultiVersion (ROMV) and 2V2PL, because they are simple enough for realization and provide the user with the advantages of most version DBMS.

In 2V2PL protocol the problem with the limitation of the number of versions is solved.

3 Basic Elements of the GPSS Simulation Models

The presented simulation model uses generated streams of transactions which imitate global transactions in DDB systems. They are all in parallel streams and their intensity λ is given in tr per sec (number of transactions per second). The modelling of the distributed transactions in DDBMS with GPSS transactions and the resolutions

2PL modeling in DDBMS with data replication are described in detail for Centralized for 2V2PL and Distributed 2V2PL protocols in [4], and [6].

The structural scheme of a modelling algorithm for distributed transactions management in two-version architecture of data in DDB is shown in fig. 1. This work suggests the modelling of distributed transactions, of which in requesting the lock of the second element is acquired higher priority and the concurrency control in GPSS World environment according to an already mentioned scheme.

3.1 Parameters of the GPSS transactions

P1 – Number of transaction. The value is a sum of System Numeric Attribute MP2 (The subtraction between the relative model time and the content of the second parameter of GPSS transaction) and the number of the site;

P2 – Number of the site, where the transaction is generated. The value is a number from 1 to <number of stream transactions>;

P\$Nel – Length of the modeled transaction. The value of that parameter in the constructed models is 1 or 2 chosen by probability defined by the function FN\$BrEl respectively 0.30 and 0.70. It is supposed that long transactions get in the system more frequently than short ones in that model;

P\$E11 – Number of the first element, which the generated transaction will read or write. The value is a random number and is uniformly distributed in the interval [1, NumEl];

P\$E12 – Number of the second element, which will be processed by the generated transaction;

P\$B11 – Type of the requested lock for the first element, which will be processed by the generated transaction;

P\$B12 – Type of the requested lock for the accessed second element;

P5 – Value 0, if the transaction is in 1st phase – occupation of the locks and value 1, if the transaction finishes its work and has to release the locks. In the future value 2 will be able to be appropriated to a transaction which will have to be restarted;

P\$Prim1 – Number of the primary sites of the first element, which the generated transaction will read or write;

P\$Prim2 – Number of the primary sites of the second element, which will be processed by the generated transaction;

P\$CHTN1, P\$CHTS1 – In the situation when P\$B11 = 1 the transaction only “reads” the element with number P\$E11. This is possible if the element is not free and the lock is permissible. According to these facts the parameters P\$CHTN1, P\$CHTS1 record accept respectively the number of the previous transact which had blocked the element and the number of the site generated it;

P\$CHTN2, P\$CHTS2 – In the situation when P\$B12 = 1 the transaction only “reads” the element with number P\$E12. This is possible if the element is not free and the lock is permissible. According to these facts the parameters P\$CHTN2, P\$CHTS2 record accept respectively the number of the previous transact which had blocked the element and the number of the site generated it;

P6 and P7 – In them there are correspondingly recorded the number of the site, where it is the nearest copy of the data element and the number of the site, where it is

the second replica of the first data element, processed by transaction. Correspondingly in parameters P8 and P9 we have the nearest copy of the second data element and the number of the recorded site, where it is the second replica of the second data element.

3.2 Basic operations

The requests for locking of the data elements l_i wait for the release of the lock manager (LM) in the queue before it QLM . They queue firstly according to a priority and secondly according to the order of getting there, thirdly according to the number of the site – generator.

The basic steps in the synthesized Primary Copy 2V2PL algorithm are:

1. Transaction coordinator TC_{P2} splits global transaction T_{P1}^{P2} in two sub-transactions, T_{P1}^{Pel1} and T_{P1}^{Pel2} , for the process of the two elements E11 and E12 (operation 1 on fig. 1). These sub-transactions T_{P1}^{Pel1} (operation 2 on fig. 2) and T_{P1}^{Pel2} (operation 5 on fig. 1) are directed at lock managers LM_{prim1} and LM_{prim2} , to get the proper locks for the elements E11 and E12.

2. If it is allowed for sub-transactions to lock the elements E11 and E12, then the correspondent records are put in the lock tables LT_{prim1} and LT_{prim2} (operations 3 and 6 on fig. 1). Moreover the sub-transaction T_{P1}^{Pel2} waits for the acknowledgment of getting the locks of E11 by T_{P1}^{Pel1} in S_{P2} . After that it is directed to the lock manager LM_{prim2} (operation 5 on fig. 1).

3. After given and granted locks (operations 4 and 7 on fig. 1), the transaction T_{P1}^{Pel1} splits in two sub-transaction $T_{P1}^{Pel1,P6}$ and $T_{P1}^{Pel1,P7}$ (operations 8 and 9 on fig. 1). If the transaction T_{P1}^{Pel1} is required to read only the element E11 then it doesn't split in two separate sub-transactions. Similarly, the transaction T_{P1}^{Pel2} splits in two-transactions $T_{P1}^{Pel2,P8}$ and $T_{P1}^{Pel2,P9}$ depending on where the copies of the data elements processed by the transaction T_{P1}^{P2} have been stored.

4. Sub-transactions $T_{P1}^{Pel1,P6}$ (and $T_{P1}^{Pel1,P7}$) and $T_{P1}^{Pel2,P8}$ (and $T_{P1}^{Pel2,P9}$) are directed to the corresponding data managers (operations 10 on fig. 1). After that they are transmitted in the network (operations 11 on fig. 1) to DM_{P6} (and DM_{P7}) and DM_{P8} (and DM_{P9}) If they have to be committed in a remote node.

5. Many operations read/write are committed in the sites-executors, where the corresponding data managers are included. The nodes PatS1 and PatS3 perform operations read/write over the first replica of E11 and E12. It is shown in fig.1. The nodes PatS2 and PatS4 perform the operations write second replica of E11 and E12 (operations 12).

6. If the sites-executors don't match to the site – initiator of T_{P1}^{P2} after the sub-transactions have finished their actions in sites S_{P6} , S_{P7} and S_{P8} , S_{P9} , they are transmitted over the communication network to the transaction manager (operations 13 and 14 in fig. 1).

7. Transaction manager of T_{P1}^{P2} sends messages, which consists of requests for releasing the elements E11 and E12 (operations 16) to the correspondent lock manager, when it receives messages for ends of sub-transactions (operations 15 on fig. 1).

8. If a new version of E11 and/or E12 is created by the transaction then the requests for end of record and release of exclusive lock wl (operations 16 on fig. 1) are getting transformed to requests for certify lock cl. Moreover higher priority is given to them.

There is existed a situation, in which the correspondent lock manager permits a record of certify lock of E11 and/or E12 (operations 17 on fig. 1) to be included in lock table, when acknowledge of certify lock (operations 18 on fig. 1) is received. In this common case, the sub-transactions T_{P1}^{Pel1} and/or T_{P1}^{Pel2} are sent through the network toward to the sites – executors to make uncommitted version of E11 and/or E12 a current one.

9. After fixing the results from sub-transaction processing in sites S_{P6} , S_{P7} and S_{P8} , S_{P9} $T_{P1}^{Pel1,P6}$ and/or $T_{P1}^{Pel1,P7}$ and $T_{P1}^{Pel2,P8}$ and/or $T_{P1}^{Pel2,P9}$, the transactions are sent through the communication network to the transaction manager (operations 20 on fig. 1).

10. Transaction manager of T_{P1}^{P2} sends messages to the corresponding lock manager when it gets the messages for the end of the sub-transactions. These messages contain requests to request a release for locks of the elements E11 and E12 (operations 21 and 22 on fig. 1).

11. Sub-transactions T_{P1}^{Pel1} and T_{P1}^{Pel2} release locks of E11 and E12. The records for these elements have been deleted from the lock tables (operations 23 and 24 on fig. 1). Afterwards, acknowledgments for the releasing of locks for E11 and E12 are sent to the manager of T_{P1}^{P2} (operations 25 and 26 on fig. 1).

12. Transaction T_{P1}^{P2} quits the system (operation 27 on fig. 1) as soon as sub-transactions T_{P1}^{Pel1} and T_{P1}^{Pel2} finish their process.

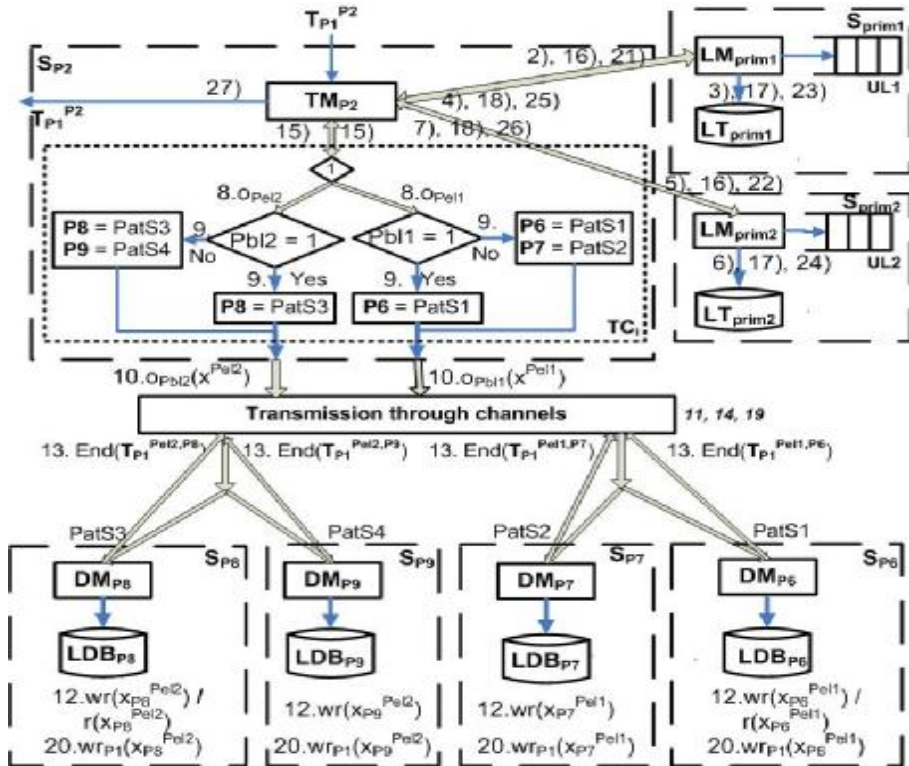


Fig. 1 Scheme of Primary copy 2V2PL

The transfer through the network to primary lock managers LM_{prim1} and LM_{prim2} and to the sites-executors, where are the data managers DM is simulated with retention.

4 Simulation

Some fragments of a program GPSS World code, specific for the distributed transactions modeling and their service for two-version architecture of data in DDB, are given here. We view a model of Primary copy 2V2PL synthesized similarly to the modeling algorithms in [4] and [6].

Example of a Computer Program from Vasileva S. (2009) GPSS World model of Primary copy 2V2PL

```

...
LockTip1 FUNCTION RN3,D3;Probability of coming up of the
.40,1/.70,2/1.0,3 ;particular type of E11 locking
LockTip2 FUNCTION RN3,D3;Probability of coming up of
.20,1/.50,2/1.0,3 ; the particular type of E12 locking
DistrS1 FUNCTION V$SiteRepl1,D6;Replication of the data
1,2/2,6/3,1/4,5/5,3/6,4 ;First node for the E11
... ;Replication of the data - Second node for the E11
Vozmozh FUNCTION RN1,D2
.05,0/1.0,1
...
PrimaS1 FUNCTION V$SiteRepl1,D6 ;Primary node of E11
1,2/2,1/3,3/4,5/5,6/6,4
PrimaS2 FUNCTION V$SiteRepl2,D6 ;Primary node of E12
1,2/2,1/3,3/4,5/5,6/6,4
...
BrE1 FUNCTION RN4,D2 ;Number of the elements accessed
.30,1/1.,2 ;transaction
; Description of the used matrices
; Description of the storages
; Description of the tables for statistic
*** Segment for transaction generation
GENERATE 60, FN$XPDIS ;Generation of
Potok1 ASSIGN 2,1 ;GPSS transactions with the
ASSIGN 1, (MP2+FN$NomSait) ;given incoming
TRANSFER ,BEGI ;interval (60 ms)
...
GENERATE 60, FN$XPDIS ;Generation of
Potok6 ASSIGN 2,6 ; GPSS transactions
ASSIGN 1, (MP2+FN$NomSait)
BEGI GATE FV FN$TraMan ;Processing in
SEIZE FN$TraMan ;transaction manager

```

```

FUNAVAIL FN$TraMan
ASSIGN Nel, FN$BrEl; Giving the number of the elements
ASSIGN E11, V$ElemN1 ; Giving the number of the E11
ASSIGN B11, FN$LockTip1 ; Lock type of E11
ASSIGN 5, 0 ; Flag of transaction committing
ASSIGN Prim1, FN$PrimaS1 ; Primary site of E11

...
SAVEVALUE BROITR+, 1 ; Count generated transaction
TABULATE TablGen
TEST E P$Nel, 1, Prebro2
SAVEVALUE BROITR1+, 1 ; accessed only one element
TRANSFER , Prebrg
Prebro2 SAVEVALUE BROITR2+, 1; accessed two elements
Prebrg ADVANCE 4
Obrab QUEUE FN$TransCor
GATE FV FN$TransCor ; Processing in Transaction
SEIZE FN$TransCor ; coordinators
...; Preparation for (not) splitting of transaction to
; sub-transactions
...; Defining values for P6,
...; and for P7,
...; and for P8, and P9
Pered2 RELEASE FN$TransCor
FAVAIL FN$TransCor
SPLIT 1, Pat01, REPLI ; Splitting of the first subT
Per2 TEST NE P$Nel, 1, PrKrai
Patlob GATHER 2
ASSEMBLE 2
SPLIT 1, Pat02, REPLI ; Splitting of the second subT
PrKrai TRANSFER , Krai
Pat01 ADVANCE MX$RAZST(P2, P$Prim1), MX$RAZDEV(P2, P$Prim1)
; Transfer in the net to the LM of Primary copy of E11.
ASSIGN 10, P$E11
ASSIGN NoE1, 1
QUEUE FN$PrimLM1 ; Waiting in front of LMprim1
SEIZE FN$PrimLM1 ; Processing in Lock manager of E11

...
TEST E P5, 0, Fiksira1
TEST E CH*E11, 0, Chakanel ; Is E11 free,
Zaema1 TEST E MX$LTA(P$E11, 1), 0, ProvR1 ; If "YES" then
MSAVEVALUE LTA, P$E11, 1, P$B11 ; the element E11
MSAVEVALUE LTA, P$E11, 2, P1 ; from lock table
MSAVEVALUE LTA, P$E11, 3, P2 ; is taken
TRANSFER , PatPot1
ProvR1 TEST E V$RAZRBL1, 1, ProvW1 ; if E11 is busy by
ASSIGN CHTN1, MX$LTA(P$E11, 2) ; a reading T and first
ASSIGN CHTS1, MX$LTA(P$E11, 3); subT wants only to read
MSAVEVALUE LTA, P$E11, 4, P1 ; it, the E11 is taken

```

```

...
ProvW1 TEST NE P$B11,1,ProvR12 ;If T doesn't read E11
TEST E MX$LTA(P$E11,1),1,ProvR12 ;and E11 and E11
TEST E MX$LTA(P$E11,5),0,Chakanel ;doesn't locked by
MSAVEVALUE LTA,P$E11,5,P1 ;rl and any other T doesn't
;create second version of E11, It's going to be
;created uncommitted version of E11
ProvR12 TEST NE MX$LTA(P$E11,1),4,Chakanel ;If another
TEST E MX$LTA(P$E11,5),0,Chet2vr1 ;T doesn't change
ASSIGN CHTN1,MX$LTA(P$E11,2) ;versions
ASSIGN CHTS1,MX$LTA(P$E11,3) ;(doesn't put cl)
...
Chakanel RELEASE P$E11 ;Waiting in the queue of
LINK P$E11,FIFO ;waiting transactions
PatPot1 RELEASE P$E11
TEST NE P$N1,1,CepiI ;The first subT and the
SPLIT 1,Patlob,REPLI ;second subT for E11
CepiI SPLIT 1,PatS1,REPLI ;is heading to the
TEST NE P$B11,1,Cepi01 ;corresponding DM
SPLIT 1,PatS2,REPLI
Cepi01 TRANSFER ,Obd1
*****Releasing of E11
Fiksiral TEST NE P5,3,Rel2Vr1 ;If T didn't create
TEST E P$B11,1,RelLock1 ;uncommitted version of E11
TEST E V$PRC1,0,ProvCh1 ;If T only reads E11
TEST NE MX$LTA(P$E11,4),0,RelLock1 ;Is T the first
;in the group adjacent reading E11 transactions
ProvCh1 TEST NE MX$LTA(P$E11,4),P1,RelLock1
RELEASE P$E11
TRANSFER ,Krail
RelLock1 MSAVEVALUE LTA,P$E11,1,0 ;Releasing E11 in
MSAVEVALUE LTA,P$E11,2,0 ;LTprim1
MSAVEVALUE LTA,P$E11,3,0
MSAVEVALUE LTA,P$E11,4,0
RELEASE P$E11
...
Fik1 LEAVE SEGM1
Krail ADVANCE
MX$RAZST(P$Prim1,P2),MX$RAZDEV(P$Prim1,P2);SubT, releasing
TRANSFER ,Krai ;lock of Element1 - to the TM
Provch01 TEST E P$NoE1,1,Provch02
TEST E P$B11,1,Izlizal
UNLINK E P$E11,Provch01,1,B11,1,Izlizal
Izlizal TEST E P$NoE1,1,Zaema02
TRANSFER ,ProvFik1 ;to the corresponding LM
Zaema02 LINK P$E11,LIFO
Rel2Vr1 ASSIGN B11,4 ;If T is accessed uncommitted
; version of E11

```

```

TEST E MX$LTA(P$E11,2),P1,ProvSp1 ;If T has wl(E11)
MSAVEVALUE LTA,P$E11,1,P$B11 ;Setting of cl(E11)
TRANSFER ,CepiVer1 ;Transfer to the DM
ProvSp1 TEST E MX$LTA(P$E11,1),0,ChakSp1 ;If E11
MSAVEVALUE LTA,P$E11,1,P$B11 ;is free
MSAVEVALUE LTA,P$E11,2,P1 ;T sets certify lock
MSAVEVALUE LTA,P$E11,3,P2 ;of E11
...
RELEASE P$E11
FAVAIL P$E11
CepiVer1 SPLIT 1,PatS1,REPLI ;Toward E11 first copy DM
SPLIT 1,PatS2,REPLI ;and DM of second copy of E11
TRANSFER ,Obd1 ;for changing versions of E11 in LDB
TRANSFER ,CepiVer1
ChakSp1 PRIORITY 5 ;Higher priority of T
RELEASE P$E11
LINK P$E11,PR
*****
...;Processing in the Primary Lock Manager of the Element2
*****
PatS1 ADVANCE MX$RAZST(P$Prim1,P6),MX$RAZDEV(P$Prim1,P6)
;subT transmission to the data manager of first replica
;of Element1
QUEUE FN$Opash1 ;read/write in the local database
ENTER FN$Opash1 ;with the first replica
DEPART FN$Opash1 ;of Element1
ADVANCE 3,1
LEAVE FN$Opash1
ADVANCE MX$RAZST(P6,P$Prim1),MX$RAZDEV(P6,P$Prim1)
TRANSFER ,Fiks1;Transmission to the primary LM of E11
...;subT transmission to the DM of second replica of E11
...;Record of the second replica of Element1
Fiks1 TEST NE P$B11,1,Obd1 ;Assemble the subT, which
GATHER 2 ;processed Element1
ASSEMBLE 2
Obd1 GATHER 2
ASSEMBLE 2
TEST E P5,0,IIvrE1 ;If T accessed committed version
ASSIGN 5,1 ;of E11, then P5=1
TRANSFER ,Zakr1
IIvrE1 TEST E P$B11,1,IIvrE1 ;If T has been read
ASSIGN 5,1 ;uncommitted version of E11, then P5=1
TRANSFER ,Zakr1
IIvrE1 TEST E P5,2,IIvrEt1 ;If T accessed uncommitted
ASSIGN 5,3 ;version of E11, P5=3
TRANSFER ,Zakr1
IIvrEt1 ASSIGN 5,1 ;If T has been changed versions of
; E11, then P5=1

```



```

Zakr1  TRANSFER ,ProvFik1 ;Toward LMprim1 for releasing
      ;of Element1 lock
.. ;subT transmission to the data managers of Element2
... ;read\write in the local databases with the replicas
      ;of Element2
... ;Transmission to the primary LM of Element2
Krai TEST NE P$Nel,1,Saber1 ;Merging of
      GATHER 3 ;successfully finished their work
      ASSEMBLE 3 ;(sub)transactions
      SAVEVALUE ZAVTR2+,1 ;Count committed transactions
      TRANSFER ,Napus ; accessed two elements
Saber1 GATHER 2
      ASSEMBLE 2
      SAVEVALUE ZAVTR1+,1 ;accessed only one element
Napus  SAVEVALUE ZAVTR+,1

      DEPART TOTALTIM
      TERMINATE 0

```

5 Simulation Results

Our research has been made for 2 replicas of the element, number of the incoming streams - 6 and number of the data elements in the global database - 50. Several parameters have been changed in order to obtain the whole picture of the states in which the chosen model went through.

The parameters and indexes of the simulations of the considered model are as follows: NumTr – general number of the generated transactions for the time of incoming modeling; FixTr – general number of the completed (committed) transactions for the same period; $X = \text{FixTr}/T_n$ – throughput of the queuing system; T_n – time interval in which the system is being watched; $P_s = \text{FixTr}/\text{NumTr}$ – probabilities for transaction service. The results are received in 6 streams of concurrent transactions with different intensity. The copies of the data elements are distributed evenly and random by 6 sites in the system. Service and rejection probabilities are calculated according to closely associated formulas and the received values are different from those received through more detailed expressions. The results of our model simulations of the model for equal intensity of 6 input flows for 2 element copies are summarized and presented graphically in fig. 2 - fig.3. The similar summarized results for different intensity of streams are shown in fig.4 – fig 7. The diagrams show the results of the conducted simulations of the presented GPSS model of primary copy 2V2PL in DDB for different input intervals of transaction coming.

Fig. 2 shows the results of throughput for simulation of primary copy two-version 2PL algorithm in same intensities of the incoming streams of global transactions. It is seen that the throughput has same behavior of increment in transition mode as the system load increases. For static mode the throughput is constant and it has max value.

The graphics in fig. 3 show the results of probability service for simulation of primary copy two-version 2PL algorithm in same intensities of the incoming streams of global transactions. The graphics in fig. 3 represent the processing in the model for different loads: min, average and max. The graphics in fig. 3 show, that the probability service has the same behavior when the monitoring time is increased and the intensity of input streams accepts tree different values. Moreover, the probability service reaches the max possible value when the static mode is used.

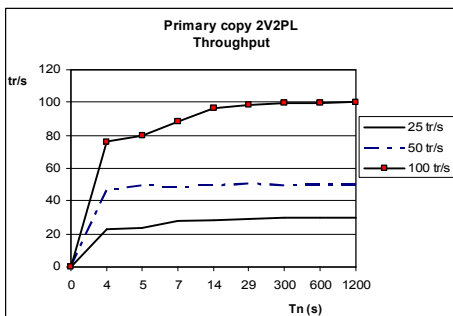


Fig.2. Throughput of the model in one and the same intensities of the incoming streams

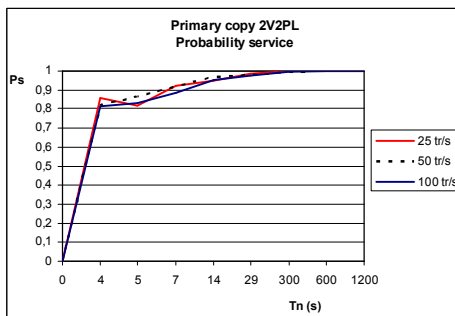


Fig.3. Probability service of the model in one and the same intensities of the incoming streams

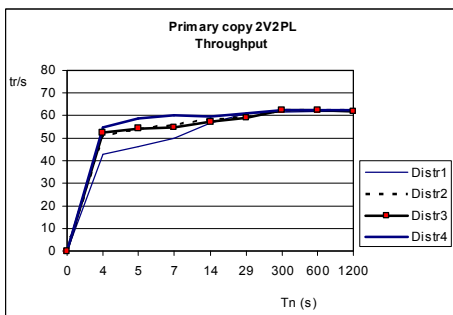


Fig.4. Throughput of the model in different intensities of the incoming streams with one and the same summary intensity 0,6 tr/s

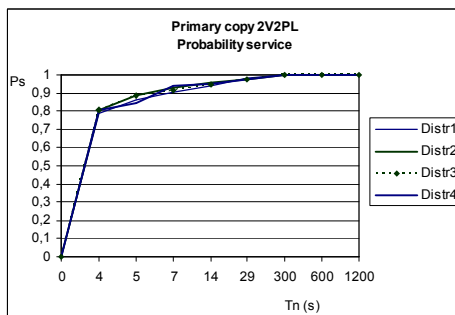


Fig.5. Probability service of the model in different intensities of the incoming streams with one and the same summary intensity 0,6 tr/s

Fig. 4 shows the results of throughput for simulation of primary copy 2V2PL algorithm in different intensity distribution of the incoming streams with one and the same summary intensity 60,87 tr/s. The diagram on fig.4 confirms the results given in the fig. 2. The intensity of the output stream with processed transactions retains its behavior when the intensity of summary input stream is constant. It is seen a sharp increment in the beginning of monitoring time interval and a constant max value for static mode.

The graphics in fig. 5 show the results of probability service of distributed transactions for simulation of Primary copy 2V2PL for different distribution of the

intensity for the six input streams with generated transactions from fig. 4. It's clear that the probability service depends only from the summary intensity of the input stream when the transition mode is running (fig. 5).

The graphics on fig. 6 demonstrate the simulation results for throughput of the system, working on the primary copy 2V2PL for different increasing intensity of the input streams. It is seen that increasing the number of the generated concurrent GPSS transactions in the model leads to an increasing of the intensity of the system output stream. As a whole it can be seen that the intensity of output system stream has being increased. Moreover, after the static mode has been running throughput accepts its max value and it doesn't change.

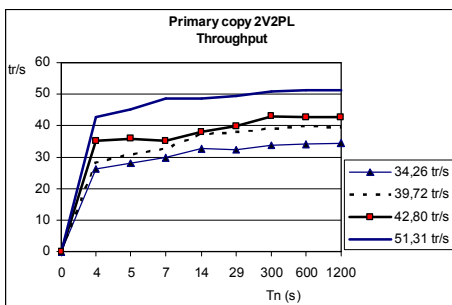


Fig.6. Throughput of the model in different increasing intensities of the incoming streams

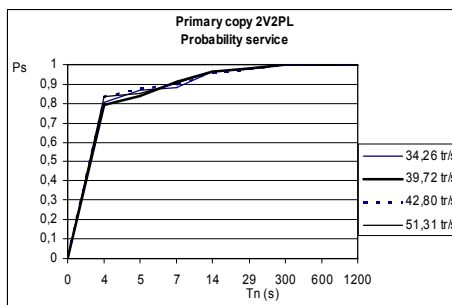


Fig.7. Probability service of the model in different increasing intensities of the incoming streams

The graphics on fig. 7 show the results for probability service of distributed transactions for the same intensities of the incoming streams as they are in fig. 6. It is seen in fig. 7 that probability of service has being increased for the forth values of the summary input stream intensity and it has its max values when static mode has been running. This situation is for the situation when an average load occurs.

6 Conclusions

It is suggested a structural scheme of modeling algorithm to control the transactions for primary copy two phase protocol in distributed managed database systems where two version architecture of the database is used.

A program code for GPSS World is developed which can be used to model the processing of distributed transactions in two-version pessimistic protocol in DDB.

Conducted simulations and the results confirm functionality of the modeling algorithm.

Simulation model allows definition the throughput ability of the distributed system, the average service time of distributed transactions and other parameters on the basis of which the efficiency of the suggested algorithm could be defined.

We continue the gathering of statistics of simulations of the modeling algorithm in different parameters of the system and the coming streams and a comparative analysis

of the results in different solutions for the data replication and the concurrency control.

References

1. Bayer, R., Heller, H., Reiser, A.: Parallelism and Recovery in database systems. In: ACM Trans. Database Syst, 5(2), pp.139--156 (1980)
2. Bernstein, Ph., Goodman, N.: Multiversion Concurrency Control – Theory and Algorithms. In: ACM Transactions on Database Systems, vol. 8, No. 4, 465--483 (1983)
3. Kuo, T., Kao, Y., Kuo, C.: Two-Version-Based Concurrency Control and Recovery in Real-Time Client/Server Databases, http://rtlab.cs.nthu.edu.tw/course/cs5425TA/2006_Spring/reading%20list/Two-Version-Based%20Concurrency%20Control%20and%20Recovery%20in%20Real-Time%20Client%20Server%20Database.pdf (2006)
4. Milev, A., Vasileva, S.: Modeling of Two-version Algorithms for Two-Phase Locking in Distributed Databases. International Conference Automatics and Informatics'09, Sofia, Bulgaria, 2009, p. I-13 -- I-16. (1999)
5. Stonebraker, M., Neuhold, E.: A Distributed Database Version of INGRES. In: Proceedings of 2nd Berkeley Workshop on Distributed Data Management and Computer Networks, pp. 9--36. Berkeley, CA (1977)
6. Vasileva, S., Noskov, Yu.: Modeling of two-version two-phase locking. Четвертая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности ИММОД-2009. Материалы конференции, Сборник докладов, Том II, с. 186--205 (2009)

Applying a Seven Streams Approach for Building a DW 2.0

Nikola Hristov¹, Kalinka Kaloyanova²

¹TechnoLogica, Sofia, Bulgaria
nhristov@technologica.com

²Faculty of Mathematics and Informatics, University of Sofia,
Sofia, Bulgaria
kkaloyanova@fmi.uni-sofia.bg

Abstract. This paper focuses on the major characteristics of the new generation data warehouse – DW 2.0. Some of them have evolved from the first-generation data warehousing, others are newly emerged. One of the basic aspects of DW 2.0 is applying the spiral methodology which adopts the seven streams approach thus attaining easier and faster DW development process. The presented case study illustrates the usage of this approach in a real environment and gives an opportunity some analyses and conclusions from applying the new methodology to be done.

Keywords: Data Warehouse, DW 2.0, ETL, Seven Streams Approach, Life-cycle of Data, Metadata, Structured and Unstructured Data

1 Introduction

The data warehousing is one of the most important and widely used practices for information management. It focuses on “integrated, granular and historical data” [1] which is used by the corporations for information and reliable reporting and analysis.

Usually data comes from different source systems and is identified with a particular time period. All source data is transformed and the result is that the final data in the data warehouse has a single physical corporate image [7]. On this ground the first-generation data warehouse works with concepts like source system, staging area, presentation area, metadata, granularity, etc. [3]. Building a successful data warehouse is also achieved through applying the ETL (Extraction, Transformation and Loading) process.

A number of driving forces like the increasing volumes of data, the raising speed and dropping unit cost of technology have shaped the evolution of the data warehousing from the first-generation data warehouses to DW 2.0 [2, 10].



2 DW 2.0

Several fundamental aspects underlie the difference between DW 2.0 and the first-generation data warehousing.

The new data warehouse technology presents the concept of “**life cycle of data**” where data passes through four different sectors:

- The Interactive Sector
- The Integrated Sector
- The Near Line Sector
- The Archival Sector

The Interactive Sector is the place where the data enters the DW 2.0 environment either through ETL [8] from an application outside DW 2.0 or as part of a transaction from an application which is located in the Interactive Sector. On leaving the Interactive Sector the data goes to the Integrated Sector. At this moment “data quality processing” like “domain checking and range checking” [2] is done. This process is also supported by ETL. Data is collected and kept here with maximum time span of three to five years. After leaving the Integrated Sector the data can go either to the Near Line Sector or to the Archival Sector.

The presence of life cycle and four sectors is an important characteristic of DW 2.0 but it is not the only one existing. Another aspect of the DW 2.0 environment is that it contains not only structured but also unstructured data. Two basic forms of unstructured data [2, 4] – “*textual*” and “*non-textual*” are supported. Textual unstructured data could be found in emails, telephone conversations etc. Non-textual unstructured data occurs as graphics and images. The integration of structured and unstructured data in DW 2.0 enables different kinds of analysis - against unstructured data or a combination of structured and unstructured data.

One of the most prominent characteristics of DW 2.0 is the spiral methodology, which adopts the so called “**seven streams approach**” [2]. Different activities are initiated and completed in each stream.

- The **Enterprise reference model stream** concerns the creation and continued maintenance of a corporate data model which consists of Entity Relationship Diagrams and background information.
- The **Enterprise knowledge coordination stream** relies on various artifacts that come out of the corporate data modeling, information factory development, and data profiling streams.
- The **Information factory development stream** is built “topic by topic”.
- The data from the source systems is analyzed “in terms of quality and completeness” in the **Data profiling and mapping stream**.
- Next in the **Data correction stream** the data is examined and decided which part of it will be corrected.
- The **Infrastructure management stream** addresses the decision what people resources and tools will be used
- Data quality monitoring and process improvement is provided by the **Total information quality management stream** [2].

The concept of metadata also evolves into DW 2.0. Here every sector has metadata of its own. The metadata for the Archival Sector is “placed directly with the archival

data” [2] while the metadata for the other sectors is kept separately from the data. The different sectors support metadata for the structured and for the unstructured data.

The DW 2.0 environment has features like life cycle, integration of structured and unstructured data, metadata, but one of its most challenging characteristics is “placement of DW 2.0 in an environment where technology could be changed as fast as the business requirements change” [4]. In order the technology to adapt to the ongoing change of the business, semantically temporal data (data that undergoes changes) should be separated from semantically static data, which does not change over time. When this happens a new snapshot of the data is created. Each snapshot is delimited by time and has *to* date and *from* date. In this way a historical record for the semantically temporal data is created.

The analysis of the characteristics of DW 2.0 presents the basic differences between the two generations data warehouses. They cover crucial aspects of the data warehousing and span from the building methodology, through the development, to the implementation of the final product.

3 Applying the Seven Streams Approach for a case study for a bank

We have applied the above described principles of DW 2.0, and designed and implemented a software system for a bank on this ground.

The bank supports two systems that manage credit and debit cards – *Internal Card Management System* and *External Card Management System*. The transactions enter the Interactive Sector through the two applications *Internal Application* and *External Application*, which are located in the Interactive Sector itself. The two card management systems use devices, for example ATM’s. The devices can be property of the bank or property of another bank. The cards with which the transactions are done are issued by the bank. Two types of transactions exist:

- *Internal* transactions - the card is issued by the bank and it is used on a device property of the bank.
- *External* transactions - the card is issued by the bank and it is used on a device property of another bank.

The transactions are monitored. When an operation is completed on a device property of the bank then the transaction enters through the *Internal Card Management System*. When the operation is completed on a device property of another bank then the transaction enters through the *External Card Management System*.

There are two cases when we speak about cards and accounts.

- *One account, many cards* - several cards can be connected to one account. For example a parent owns the account and issues cards to ones own children.
- *One card, many accounts* - several accounts can be connected to one card. For example payment account, deposit account, saving account etc. There is always one default account connected to a card when it has been issued.

Based on the two source systems a data warehouse is created. It integrates the information from the transactions achieving fast data access, effective reporting and analysis.

According to the DW 2.0 theory we apply the **spiral methodology** which adopts the **seven streams approach for building our system**. We have organized and described the activities done for the different streams while creating the case study model (Table 1).

Table 1. Activities applied and achieved in the different streams

	Streams	Activities
1.	Enterprise reference model stream	Basic concepts like transaction, card, card account, currency, terminal type are defined
2.	Enterprise knowledge coordination stream	Data models are created for the structured and the unstructured data
3.	Information factory development stream	The answers to the two business questions “What is the profit from fees” and “What is the number of transactions made by physical and corporate persons” are found.
4.	Data profiling and mapping stream	The examined attributes <i>acc_no</i> and <i>ap_code</i> from the source systems are mapped to load the <i>acc_number</i> field in the target table.
5.	Data correction(cleansing) stream	The number of the merchant comes with some leading and trailing spaces. They are cleansed .
6.	Data infrastructure management stream	The tools “Oracle Warehouse Builder”, “Oracle BI Answers”, “Oracle BI Publisher”, “Oracle Text” are chosen .
7.	Total information quality management stream	Data quality is monitored – misspelled customers’ names, wrong addresses exist.

Starting with **the Enterprise reference model stream** our first step is to create a data model that describes the business. The bank data is analyzed top down from the major business subject area (the top) which is the card transaction subject area to the entity attributes (the bottom). Basic concepts in this subject area are **transaction, card, card account, currency, terminal type** and each one of them is part of the conceptual model of the upper mentioned subject area.

On the basis of the data model and the data coming from the two card management systems a corporate data view is created. The data arrives from the *External Card Management System* and the *Internal Card Management System* at an hourly base. The currency and the date format are different in the two source systems. The corporate data view is in BGN with date format of mm-dd-yyyy and at an hourly level.

The Enterprise knowledge coordination stream relies on the artifacts that come out of the three data discovery streams [2] - the Enterprise reference model stream, the Information factory development stream, and the Data profiling stream.

The bank needs answers to some topical questions such as “What is the profit of the bank from the fees?” and “What is the number of transactions made by physical persons or corporations”. These questions concern **the Information factory**

development stream. The information required to answer these questions is analyzed and the data model structure is designed.

The corporate data model is located between the interactive and the integrated layers. A major transformation of the data is made as it passes from the Interactive to the Integrated Sector. The Star Schema Model [9] is applied to the structured data in the Integrated Sector. As the data passes from the Integrated to the Near Line Sector there is no change of the data model. The Star Schema Model is preserved when data passes from the Near Line Sector to the Archival Sector.

Besides the structured data there is some amount of unstructured data that needs to be taken care of [6]. This unstructured data comes in the form of e-mails exchanged with the people from the business, notes concerning different topics such as interests, accounting or magazines. The unstructured data is gathered and after that organized into themes. Each text has major themes. For each piece of text themes and the relationships between them are defined. On this basis the internal data model of the unstructured data is created [10].

After the analysis and design phases the implementation starts.

A database is created for the purposes of the bank. It contains the data warehouse where separate schemas are created for each environment of the DW 2.0. At this stage of the implementation the schemas created are two - one for the interactive and one for the integrated environment. The data in the Near Line Sector will be stored on cartridges that will be handled automatically and the environment will be created at a later point when the data in the integrated environment grows old or the probability of access diminishes. The environment of the Archival Sector will be built even later and the data will be stored on magnetic tapes.

The Interactive storage environment of the DW 2.0 for the bank is created first. The transactions enter the Interactive Sector through the two applications *Internal Application* and *External Application* which are located in it. In this sector the data is very recent – just a few seconds old. When the data becomes 24 hours old or older, up to one month, the integrated storage environment of DW 2.0 for the bank is built.

The unstructured data that is to be loaded in the DW 2.0 is in an electronic format. Before a text enters the DW 2.0 environment it is put under transformation (removal of stop words like the, and, that etc.) through an ETL process [5].

The integrated storage environment includes the building of fact tables and the dimension tables in the tablespace allocated for this purpose. The needed indexes and sequences are generated in the corresponding tablespaces. Date partitions are created for the fact table.

The fact table is created with [11, 12]:

- the “parallel” clause (it parallelizes the creation of the table and sets parallelism for “insert”, “update”, “delete” and “merge” after the creation of the table);
- “compress for all operations” clause with which the Oracle Database tries to compress data during “insert”, “update”, “delete” operations on the table;
- “partition” clause which divides the table into day partitions;
- date timestamp.

The dimension tables are created with primary keys and unique keys. The records in the dimensions have validity in time. They have *date from* and *date to* attributes.

When the mapping process is finished transformations are created. They describe where the data is extracted from, what kind of transformations are going to be applied on the attributes and where the transformed data will be loaded. Metadata describing objects like tables, indexes, attributes etc. will also be stored in “Oracle Warehouse Builder”. A separate schema is created in the database for the repository of the “Oracle Warehouse Builder”. All the metadata is organized in system tables. When there is need the metadata to be analyzed, queries can be done against these tables.

The Local metadata for the Business Intelligence is stored inside the BI Metadata Repository. It consists of three layers of information – the Physical Layer (where the tables from the source are imported and described), the Business Layer (where the physical layer is organized into logical tables) and The Presentation Layer (this is the layer which presents the data to the end user).

The combination of all local metadata coming from the different tools forms Enterprise metadata. The union of all the Enterprise metadata which is situated above the sectors forms the Enterprise metadata repository. It is physically separated from the sectors i.e. there is a different schema created in the database for the Enterprise metadata.

The Near Line Sector and the Archival Sector and their metadata will be built at a later time when the need of their existence appears.

The Local metadata is related to the Enterprise metadata. The global term for the bank is “fee” and is described on Enterprise level. On this level the local usage of the term and the pointer to the local system where the term is found are also described. The local term in the “business layer” of the BI Administration tool is the same – “fee” and in the “Oracle Warehouse Builder” is “charge”.

When the data grows older (about three or four years old), then the near line storage environment will be built. The data model here will be the same as the data model found in the integrated storage environment which is a Star Schema model. When the data grows older than five years we will need the archival storage environment. The data model here will still be Star Schema Model.

In a DW 2.0 environment the technology must change as fast as the business changes. [2] This is achieved by separating semantically static from semantically temporal data. In our bank example we achieve this separation by using one fact table and several dimension tables which track historical changes.

Reporting is done as a last step of building the DW 2.0 for the bank. The tools “Oracle BI Answers” and “Oracle BI Publisher” are used for the reporting [13].

The answers to the two topical questions that the business asked are found in the reports created with the BI tools. The report “profit_from_fees”, shown on Fig. 2 gives information about the profit of the bank from fees in “bgn” currency. It is created in “Oracle BI Answers” and the result is placed on a dashboard. Dashboard is the page where people can view regularly their reports.

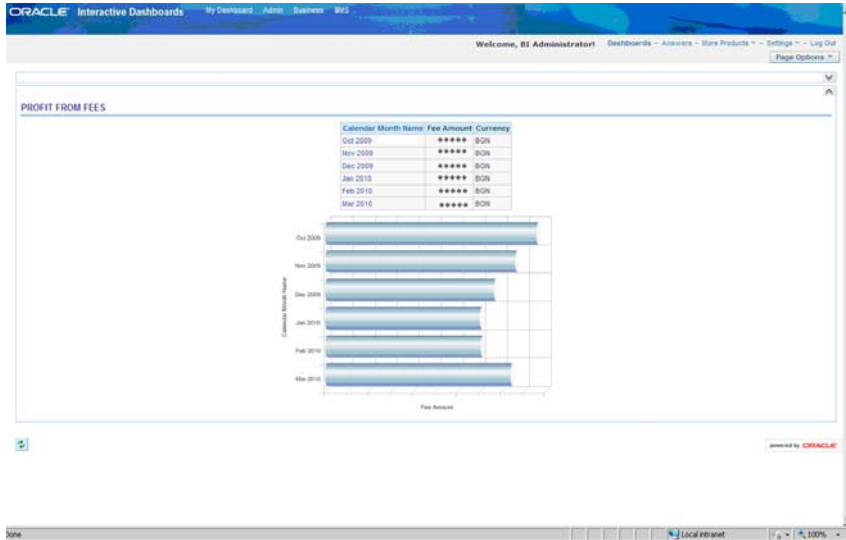


Fig. 2. Profit from fees

In the Data profiling stream data profiling and mapping is done. The actual detailed data about the transactions from the two online transaction systems is examined and analyzed in terms of quality and completeness.

In the Data correction stream the data that needs to be corrected or purged is identified going attribute by attribute. In our case the number of the merchant is a “Varchar” field and it comes with some spacing at the beginning or at the end. The correction rules to remove these redundant space characters are chosen and applied.

The tools for development are selected during the activities in the Infrastructure management stream. We use the already mentioned “Oracle Warehouse Builder”, “Oracle Text”, “Oracle BI Answers” and “Oracle BI Publisher” for the model of the bank.

Finally, specific data elements in the environment are examined in the Total information quality management stream and their quality is monitored and reported over time.

4 Conclusion

In this paper we present the basic concepts and aspects of the new generation data warehousing. The advantages and challenges of DW 2.0 have been pointed out by building a case study for a bank which applies the new methodology.

We have combined the spiral methodology with the seven streams approach, paid enough attention to the data model, coordinated the processes of mapping and correction of data and monitored the whole development process thus achieving easier and faster construction of the data warehouse, than the construction of a first-generation data warehouse.

A valuable peculiarity when applying this approach is that the development phases are initiated in parallel and that the requirements can be gathered during the whole data warehouse development process. The combined support of structured and unstructured data, as well as the improved model for metadata management complement the advantages of the new methodology and contribute for the fast discovering of the needed data and extraction of information from it. This facilitates the decision making process, the results from which are of great importance for the future development of one company.

What is attained by developing DW 2.0 is answer to the topical business questions, fast and effective reporting, integrity of structured and unstructured data, maintenance and storage of metadata, fast data access, flexibility concerning the changing business environment.

Acknowledgments. This paper is supported by the Sofia University SRF under Contract 163/2010.

References

1. Inmon, William: Building the Data Warehouse, Fourth Edition, Wiley Publishing, Inc., Indianapolis, Indiana (2005)
2. Inmon, William and Strauss, Derek and Neushloss, Genia: DW 2.0: The Architecture for the Next Generation of Data Warehousing, Elsevier Inc. (2008)
3. Kimball, Ralph and Ross, Margy: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, Second Edition, John Wiley and Sons (2002)
4. Inmon, Bill: Different Kinds of Data Warehouses.
<http://www.b-eye-network.com/view/10416>, (2009)
5. Santos, Ricardo Jorge, Bernardino, Jorge: Optimizing Data Warehouse Loading Procedures for Enabling Useful-Time Data Warehousing, (2009)
6. Russom, Philip: Complex Data: A New Challenge for Data Integration, (2007)
7. Chaudhuri, Surajit and Dayal, Umeshwar: An Overview of Data Warehousing and OLAP Technology, ACM New York, NY, USA (1997)
8. Rahm, Erhard and Do, H. H.: Data Cleaning: Problems and Current Approaches, (2000)
9. Imhoff, Claudia and Galleo, Nicholas and Geiger, Jonathan: Mastering Data Warehouse Design: Relational and Dimensional Techniques, Wiley Publishing, Inc., Indianapolis, Indiana (2003)
10. Kozielski, Stanislaw and Wrembel, Robert (Eds.): New Trends in Data Warehousing and Data Analysis, Springer Science+Business Media, LLC, (2009)
11. Oracle® Database Administrator's Guide 11g Release 2 (11.2).
http://download.oracle.com/docs/cd/E11882_01/server.112/e10595/manproc006.htm#ADMIN11186
12. Oracle® Database SQL Language Reference 11g Release 2 (11.2).
http://download.oracle.com/docs/cd/E11882_01/server.112/e10592/wnsqli.htm#SQLRF50926
13. Oracle® Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide.
http://download.oracle.com/docs/cd/E12096_01/books/PDF/AnyUser.pdf

Distributed Systems and Grid Technologies

Ontology Design and Development for Grid Services

Xhemal Zenuni, Florije Ismaili, Bujar Raufi

Faculty of Contemporary Sciences and Technologies, SEEU, Ilindenska bb, 1200 Tetovo,
FYR of Macedonia
{xh.zenuni, f.isamili, b.raufi}@seeu.edu.mk

Abstract. This paper focuses on the definition of an ontology to semantically describe Grid Services as specified by Web Services Resource Framework (WSRF). The proposed semantic framework model integrates both aspects of Grid Services profile: the functional aspects and Quality of Service (QoS) features. We have developed a model that is enough generic to be applicable across multiple application domains, that is rich to express multidimensional aspects of different service facets, and sufficiently robust to be applicable when integrated with the proposed vocabulary. The ontology is aimed to provide an enhanced support to efficient Grid Services discovery and composition objectives based on complex users' requirements, to maximize the utilization of Grid infrastructure and to contribute to Semantic Grid vision.

Keywords: Ontology, Grid Services, WSRF, Semantic Grid

1 Introduction

Grid computing continues to draw a great deal of interest, both in academia and industry, as it provides a novel way of integrating distributed and heterogeneous computer resources residing across multiple organizational pools. Currently, there is a multitude of proposed Grid architectures and applications, yet the latest generation of Grid [1] is moving toward Service Oriented Architecture (SOA) principles to Grid computing, as well suited model to address most of the emerged critical issues that arise in Grid environment, such as heterogeneity, interoperability, scalability or enforcement of local policies of resource usage among others.

The adoption of Web Service Resource Framework (WSRF) [2] formally leveraged SOA principles to Grid computing, and placed Web Services technologies as a foundational framework on which current and future Grid application should be developed. These specifications extended traditional Web Services with features such as the notion of state, enhanced addressing, lifetime management, and grouping in order to make them suitable for sharing and managing resources in Grid. The extended Web Services with these new features, in Grid computing community and literature are known as Grid Services. Consequently in this model, Grid resources are encoded, exposed and utilized through Grid services, which provide uniform access to the underlying shared Grid resources.



With the appearance of WSRF, some of the drawbacks noticed in previous generations of Grid computing architectures, such as scalability, interoperability and the enforcement of local policies on Grid resource usage were to great extent eliminated, however new challenges were introduced. With this model of Grid computing, especially on the Grid designed to operate on Internet large scale, the number of offered services can be large and diverse. The following issues appear emergent and need to be addressed in order to support the success of Grid:

- Efficient discovery of available services to build Grid applications, that fits to complex both functional and QoS aware user's/applications' requirements and needs
- Aggregation of complex services, and generation of workflow from individual services, if grid applications can not be build by a single service
- Automation in discovery, workflow generation, invocation and management of Grid services.

Grid Services inherited from Web Services the UDDI [3], as the only accepted yellow page standard directory for Grid Services registration and discovery. However, UDDI lacks structural and semantic expressiveness in describing different functional and especially QoS quality of services. Therefore, discovery is based on simple keyword matching, resulting in high recalls and low precisions.

Lately, the potential of Semantic Web technologies has been recognized [4] as an alternative solution to overcome the limitations noticed with UDDI; however it is commonly agreed that Semantic Grid still has a long way to go to become reality. Most of the work on integrating semantic technologies for service discovery and service composition is based on OWL-S [5], and limitations have been identified at least in two directions: the semantic description of the Grid "resource" is absent, and secondly and most importantly, the QoS features are not represented. QoS parameters are important discriminating and ranking factor if there are more functionally equivalent discovered services.

This paper formulates an ontology model to semantically describe Grid Services features, including their functional aspect and Quality of Services parameters. We provide an integrated solution by focusing on the development of a rich model that is able to represent different features of important parameters for Grid environment. In addition, a rich vocabulary is proposed, thus providing a robust framework and fundamental building block toward the support of Semantic Grid vision.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents some of the design principles that guided our work. Section 4 explains the details of the proposed generic ontology, while explaining the proposed vocabulary as well. And finally section 5 draws conclusions and gives directions for the future work.

2 Related Work

A detailed review of the relevant literature and work for Grid Services ontology indicates that the research works are quite fragmented and generic, mostly focused on Web Services, and not fully adequate to be applicable in the third generation of Grid.

S-OGSA [6] is proposed as unified platform for integrating semantic metadata with Grid computing infrastructure, including a formal framework and guidelines to the development of semantic Grid applications. However, S-OGSA model is generic and allows the existence of different kinds of knowledge to be attached to any resource or Grid component.

OWL-S [7] is a standard proposed by W3C consortium to describe different type of services. It semantically describes services in terms inputs, outputs, pre-conditions and effects model, so there is no doubt on their meaning. However, Grid services differ from Web services by the possibility of having state (acting upon resource). OWL-S is limited in describing the WS-Resource, but even more importantly, it mainly focuses on the description of the functional part of services, and QoS support is almost non-existent.

A more prominent work has been conducted to proposing and defining different ontologies for QoS. OWL-Q [8] extends OWL-S for describing QoS aspects of Web Services. The ontology is quite modular and extensible, and provides detailed specification of different information, such as metrics, units, value types and relationships. However, it has weak support on expressing the impact of the parameters, their priority and a concrete vocabulary is absent.

QoSOnt [9] it is another ontology proposal in the plethora of QoS ontology models. It represents a minimal set of generic QoS concepts, and as such it has many shortcomings in terms of not providing semantic information about the direct impact of the parameters, their dynamism, units, relationships, etc.

The QoS ontology language [10] proposal is focused on the development of an QoS ontology and contains necessary classes to describe different aspects of QoS service attributes, such as metrics, the impact, nature, and so on. Moreover, a concrete vocabulary is also developed. Although it provides a quite complete list of QoS qualities, it has weak support in defining metrics, value types and it does not facilitate the conversion between different QoS parameters.

For a more complete survey on QoS ontology proposals, the paper by Tran and TSUJI [11] may serve as a starting point.

3 Requirements

The work here has been driven by four major objectives: correctness, completeness, robustness and compliance. We aim to develop an ontology that will reflect the specifics and requirements of Grid Services and its operational environment, including a description of a service “resource” among others. We also intend to provide a complete ontology that will include semantic information for both aspects of Grid Services: it should describe the functionality of provided services and its quality features. In addition, we want operational ontology representation, thus going

beyond the creation of a generic model only. In this light we develop a concrete vocabulary as represented in section 3.3. And the compliance with existing standards is an advantage that ensures existence and usage of available tools, and provides better chances for re-use and acceptance of the ontology.

On the other hand, robustness can be achieved by creating an ontology founded on clear design principles. This is valid especially for QoS part. Delivering QoS in Grid applications is not only competitive advantage, but a necessity that must be addressed seriously and created upon well defined requirements. After reviewing the related work, the key requirements, especially for vigorous QoS delivery are highlighted in what follows, and some of them are motivated from [11]:

- *Modularity*: allowing specification for the whole and part of Grid Services. Thus, different parts can be re-used independently for different situations.
- *Scalability*: the ontology should be easily extensible if new elements are identified and need to be added at some later time. This will allow the reusability of the fundamental construct.
- *Refinement support*: the ontology should allow the refinement of basic constructs.
- *Impact Factor*: to give information how and to what extend the given QoS parameter influences the overall quality of the service, such as positively, negatively, none and so on. In many situations, larger values not necessarily means better. For example, the cost factor is one of such features.
- *Dynamism*: to specify if the value of a given QoS is static or it is periodically updated. In some cases, the values of QoS features are static, such as when providing information about the supported protocols for description or invocation, but there are many cases where such properties can be dynamically changed.
- *Validity Time*: to denote how long the value of the QoS is valid, in order to be correctly evaluated by service requestor.
- *Relationship*: to specify the relationships of different QoS facets in the model. For example, the response time is connected with request time, execution time and reply time.
- *Units*: to allow the existence of several types of units, and to support their equivalence.
- *Transformations*: since the ontology can host different metrics, type units and values, a mechanism to do correct transformations is required.
- *Metrics*: to describe several characteristics of metrics, such as being simple or complex, to denote the relationships with other metrics and so on.

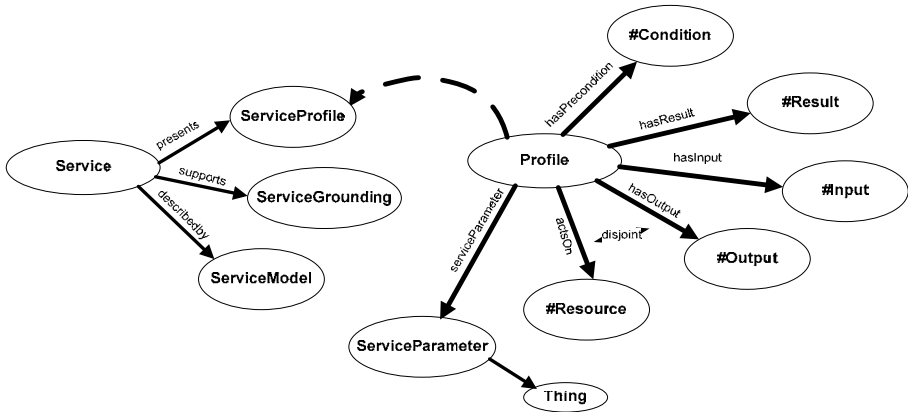
4 Building the Ontology for Grid Services

Based on the requirements set in the previous sections, we have developed ontology for description of Grid Services. In subsection 4.1 we explain the model to describe the functional aspect of Grid Services. Section 4.2 present the QoS ontology model, and Section 4.3 proposes a taxonomy and concrete list of QoS.

4.1 The Functional Features of Grid Services

OWL-S [7] provides a model to describe functional aspects of services in terms of input, output, pre-conditions and effects. The model is well grounded for Grid services as well, by the difference of the output parameter. Grid services usually acts upon a resource, and it does not provide simple output in this case. The difference is both technical and semantic. Therefore, the semantic description of the resource must be included into the ontology (resource class is disjoint with the output class), just like in the Figure 1.

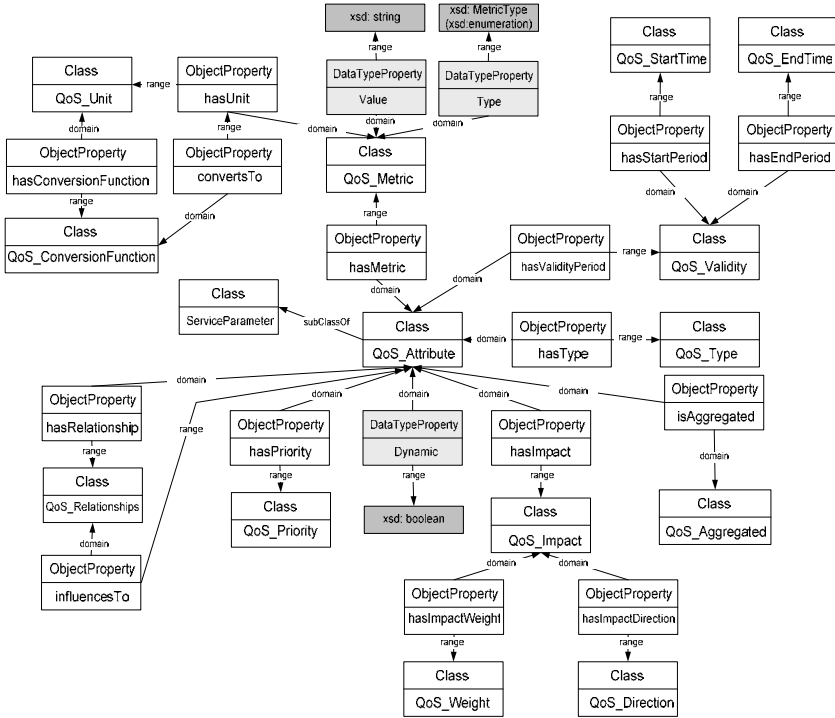
Fig. 1. The inclusion of Grid resource into Service Profile



4.2 The Ontology for QoS Facets

The proposed QoS ontology model provides a generic model to describe arbitrary QoS parameters, and it is rich in capturing and describing different facets of importance in QoS nature. The list of most important elements such as classes, their relationships and so on can be seen in Figure 2, and explanations for them are provided in subsequent paragraphs. A few other elements such as DataType property in QoS_Metric class, Ratio, Formula, System of Units, Reference ObjectProperty are defined but they are not directly visible in Figure 2.

Fig. 2. The QoS ontology asserted model (most important elements)



Each individual QoS parameter, such as Jitter or latency is represented by its *QoSAttribute* basic class. This class is subclass of *ServiceParameter* from *ServiceProfile*, which provides the link with the functional part of Grid Services. In addition, a *QoSAttribute* can be linked for specific element from *Profile* such as specific *Resource*, through *Reference* ObjectProperty that ranges to *ServiceElement* class.

The QoS can be dynamic or static, but not both at the same time. Therefore, a *DataTypeProperty* *Dynamic*, which takes Boolean value, denotes whether the given parameter is dynamic or not. Each QoS parameter can have metric, and it can be measurable and non-measurable. If it is non – measurable, it has *Value* and *ValueType*, and no *QoS_Unit*. On the other side, if it is measurable, it has specific *Value*, a *MetricType* such as being *simple* or *complex* and *ValueType*. Moreover, it has *QoS_Unit* which can be transformed to other units through *QoS_ConversionFunction* class. This class have *ratio* and specific *formula* which can be used to make necessary *System of Units*.

The specific QoS parameter contributes to the overall quality as perceived by the user. The QoS have *QoS_Direction* impact, such as positively or negatively, and its weight. Not all QoS have equal impacts to service quality, therefore the weight denotes if this impact is strong or weak.

Also, the QoS parameter can be aggregated (*QoS_Aggregated*). For example, the response time is connected with request time, execution time and reply time. This relationship between different QoS is expressed through *QoS_Relation*.

As noticed during requirement analysis, each QoS has its validity period and priority. The asserted classes for these facets are denoted as *QoS_Validity*(with two subclasses *QoS_StartTime* and *QoS_EndTime*) and *QoS_Priority*, which can be set as high priority, medium or low priority.

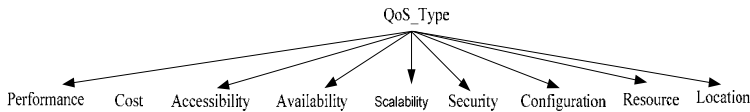
And finally, a concrete fundamental taxonomy for QoS is provided through *QoS_Type* class and appropriate subclasses.

Most of the QoS facets are represented in form of *classes*. This is done with the aim to provide an extensible and generic ontology scheme.

4.3 QoS Vocabulary for Grid Services

Delivering reliable QoS over network, especially over Internet is challenging task. Here we are more concentrated on building a concrete taxonomy and list of QoS that are important in Grid. Figure 3 depicts a fragment of QoS taxonomy and subsequent paragraphs provide more detailed explanations.

Fig. 3. A fragment of QoS taxonomy



1. *Performance*: defines the key factors which determine the speed at which the given service operates. Such factors include **latency**, **throughput**, **response time**, and **Jitter** among others. It must be noted that nature of the infrastructure, especially on Internet scale can be very dynamic; therefore these factors are expressed in terms of min/max expressions.
2. *Availability*: represents a class of parameters describing presence of the service and its readiness to be used. Its parameters indicate the probability that a service is available at given time, or if the service is down, then how much time it takes to be repaired. While for **availability time** larger values means better, for **time-to-repair (TTR)** factor smaller values are more desirable.
3. *Scalability*: specifies the probabilistic of the success rate with which the service can serve clients.
4. *Accessibility*: Higher degree of **accessibility** means that the service can serve more clients at any given point of time, and that clients can use the service relatively easily.
5. *Reliability*: denoting information about the reliability of the service, such what is the **mean time between failures (MTBF)**, or what is its **error rate**.
6. *Security*: involves aspects related to security levels that must be maintained during the communication of the users' applications and the provided services. These security concerns are related to both data and users, and involve aspects such as

message *confidentiality*, *integrity*, *authentication*, *authorization*, *digital certificates* and so on.

7. *Configuration*: describes the compliance with different *supported standards* for proper invocation and usage, such as the supported versions of WSDL description used, or SOAP version needed for successful message exchange, and so on.
8. *Resource*: this class of properties is related to the resource on which the Grid Service acts, such as the *capacity* of the resource, if it persistent resource and can survive service container restarts, or if it is in - memory *resource type* only, and so on.
9. *Location*: parameters providing *location* details where the service resides, including the *country name* and details about the *business entity* that provides the service.
10. *Cost*: it is likely to expect that people or institutions will be willing and motivated to share resources if they are financially compensated by service requestors. This is likely to be true especially in open environments, like Internet scale grids. The price can significantly influence the choice of the clients, if more than one service meets the requirements. Therefore, it is important to have a class of parameters in ontology that will indicate the pricing of the Grid service usage, such as *amount*, *currency*, *unit of charge* or *billing information*.

The ontology is developed in OWL, using Protégé [12] as development tool. An instantiation (snippet) of Grid Service Profile, with arbitrary vales and in human readable format may look like in Table 1.

Table 1. A snippet to an arbitrary human readable Grid Service profile with QoS

hasInput	actsOn	QoS_Attribute	QoS_Metric (Value)	QoS_Unit	QoS_Type
price	auction	Throughput	500	Kbps	Performance
		Response Time	150	msec	Performance
		Jitter	2	msec	Performance
		Latency	300	msec	Performance
		AvaliabilityTime	98	%	Availability
		TTR	2	sec	Availability
		Standards	SOAP 1.2	NULL	Configuratio n
		Standards	WSDL 1.1	NULL	Configuratio n
		MTBF	100.000	minutes	Reliability
		Accessibility	97	%	Accessibility
		Capacity	200	MB	Resource
		MemoryType	Persistent	NULL	Resource
		Confidentiality	DC	NULL	Security
		Price	5	\$ per month	Cost

5 Conclusions and Future Work

Up to date and to our best knowledge, grid community failed to agree on common ontology standard for grid computing. Current efforts mainly focus on defining complex ontological frameworks that will try to capture core concepts, classes, attributes and relations in grid, which will be applicable in any type of grid. They are complex, generic and do not provide any direct semantic support for the latest generation of grid.

Web service community has done major steps forward in this direction, by proposing several upper domain ontologies in describing web service functional and non – functional aspects. However, the research efforts in formulating ontologies have been quite fragmented and incomplete, not fully addressing the requirements and features of Grid Services, and in most cases not directly applicable.

This paper is concerned with the creation of novel, rich and extensible ontology that provides an integrative solution which is flexible for describing Grid Services functionalities and features, as the latest technology framework for Grid computing. It allows easily integrating and describing arbitrary QoS parameters, as an important provision in this model. Combined with the proposed vocabulary, it formulates robust semantic framework that aims to combine the advantages of semantic web technologies in one hand, and to be applicable model on the other side.

Our future work will be mainly focused on the following directions:

- To do further refinements with the vocabulary: by identifying and adding new concepts in the proposed taxonomy, and to decompose those that are generic at this point
- To integrate this ontology to our framework [13], where we expect its contribution to be further validated
- To build efficient service discovery and service composition techniques on top of it

References

1. D. D. Roure, M. A. Baker, N. R. Jennings, and N. R. Shadbolt. *Grid Computing: Making the Global Infrastructure a Reality*, chapter The Evolution of the Grid, pages 65–100. John Wiley and Sons Ltd. Publishing, New York, 2003.
2. Web Services Resource Framework (WSRF) - Primer v1.2, May 2006. Available online at: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf (Accessed May, 2010).
3. Universal Description, Discovery and Integration (UDDI). Available online at: <http://uddi.xml.org/> (Accessed May, 2010).
4. D. D. Roure, N. R. Jennings, and N. R. Shadbolt. The semantic grid: Past, present, and future. In *Proceedings of the IEEE*, volume 93, pages 669–681, March 2005.
5. D. Martin et al. Bringing Semantics to Web Services: The OWL-S Approach. In *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, 2004.

6. O. Chorco et al. An overview of S-OGSA: a Reference Semantic Grid Architecture. Available online at: http://www.dia.fi.upm.es/~ocorcho/documents/JWS2006_CorchoEtAl.pdf (Accessed May, 2010).
7. OWL-S: Semantic Markup for Web Services. Available online at: <http://www.w3.org/Submission/OWL-S/> (Accessed May, 2010).
8. K. Kritikos and D. Plexousakis. *Semantic QoS Metric Matching*. In Proc. of the European Conference on Web Services (ECWS2006), IEEE Computer Society, pp. 265–274, 2006.
9. G. Dobson, R. Lock, and I. Sommerville. *QoSOnt: a QoS Ontology for Service-Centric Systems*. In Proc. of the 2005 Euromicro SEAA, 2005.
10. I. V. Papaioannou, D. T. Tsesmetzis, I. G. Roussaki, and M. E. Anagnostou. *A QoS Ontology Language for Web Services*. In Proc. of the 20th International Conference on Advanced Information Networking and Applications (AINA2006), IEEE Computer Society, pp.18-25, 2006.
11. V. Tran and H. Tsuji. *Semantics in QoS for Web Services: A Survey*. Available online at: <http://sigswow.org/papers/SIG-SWO-A801/SIG-SWO-A801-02.pdf> (Accessed May, 2010).
12. The Protégé Ontology Editor and Knowledge Acquisition System. Available online at: <http://protege.stanford.edu/> (Accessed May, 2010).
13. Xh. Zenuni and I. Momtchev. *A Multiagent Framework for Grid Computing*. International Journal on Information Technologies and Security, Issue no.3, 2009. ISSN:1313-251.

Early Evaluation of NPB UA Benchmark Scaling to Thousands of Blue Gene/P Cores Using PGAS-like OpenMP Extension

Anton Korzh¹, Oxana Dzhosan²

¹ NICEVT,

Varshavskoe shosse125, 117587 Moscow, Russia

² Lomonosov Moscow State University,

Leninskie Gory, MSU, 2-nd educational building, 119991 Moscow, Russia

anton@korzh.ru, oxanad@mail.ru

Abstract. This article demonstrates parallel implementation of Unstructured Adaptive – one of the NASA Parallel Benchmarks – in PGAS paradigm extending OpenMP paradigm for distributed memory supercomputers. Details of modified OpenMP benchmark version to the PGAS variant are discussed as well as details of PGAS-like SHMEM library implementation for IBM Blue Gene/P through DCMF messaging layer. Results obtained on up to 4096 cores of Blue Gene/P system installed in Lomonosov Moscow State University are provided.

Keywords: NPB UA Benchmark, PGAS paradigm, Unstructured Adaptive, OpenMP extension, Blue Gene/P, parallel programming

1 Introduction

Expected in 2010 successor to Cray XT family, a new generation Cray Baker systems will be essentially traditional MPP systems, having up to thousands of nodes with two AMD Opteron processors, connected with a 3D-torus proprietary interconnect. The main difference to old generation will be the Gemini-codenamed network. One of the most important differences of Gemini in comparison to Seastar2+ chips will be not the increased bandwidth and reduced average and maximal latency but efficient hardware support of PGAS paradigm. That includes mostly remote memory accesses and extremely high Message Rate – up to tenth of millions packets per second, that is very important for very short messages. Another DARPA HPCS funded project by IBM is a Bluewaters system expected in 2011. It will use superscalar multithreaded POWER7 processors connected by extremely quick interconnect having up to 400 Gbit/s bandwidth per node. Further details remain unknown but it's stated that PGAS programming concept will be also supported.

More than 99 percent of currently running HPC applications are written using MPI library (and mostly in C and Fortran programming languages). Experience shows that users do not want to switch to another paradigm or programming language. Scientific



uses are remaining very conservative in this area. The question is for what kind of problems and mathematical methods it can be shown the benefit from PGAS programming concept in comparison with MPI and OpenMP. Supposedly these problems are the ones having essentially irregular data access pattern, e.g. very low granularity of data accesses, low spatial and temporal locality of data accesses, unpredictable dynamical accesses pattern changed. Good example of such problem is an UA benchmark from famous NASA benchmark collection NAS Parallel Benchmark, added lately to version 3.1 of NPB suite. It was developed especially to provide a standardized method for gauging the performance of parallel computer systems when running scientific applications whose memory access patterns are irregular and unpredictable. Therefore it was used by authors to demonstrate the benefits of PGAS programming concept.

Suggested benchmark solves numerically the Dirichlet boundary problem for the heat transfer PDE on a three-dimensional cubical domain using adaptive unstructured mesh[2,3]. The most important fact for the data access pattern is that mesh is unstructured and every 5th step it is being adapted - some elements close to heat source are being refined the other can be merged if finer resolution is no longer needed on that subdomain.

There are known two implementations of the benchmark: one sequential and one parallel, using OpenMP directives. Both are written in Fortran 77 and are very similar. There are no any MPI-version available, therefore no possibility to run the UA benchmark on distributed memory systems and Infiniband-based clusters. Authors in [3] mentioned that they have created a multithreaded version for a shared-memory Cray MTA2 system.

The original NPB-OMP version benchmark code is written in Fortran77, giving slightly less than 300KB of code, including comments. We have modified this version, mostly dividing data and inserting communication calls in PGAS programming concept. Then a simple realization of PGAS-like library was implemented by author, on IBM Blue Gene/P system. As communication layer was used not the MPI library, but the DCMF[1], available for BG/P. The time for adapting program to PGAS programming model was approximately two weeks, what demonstrates high productivity of PGAS programming concept. The program can be run not only on Blue Gene/P system, but also on any system supporting remote memory writes. After some modification the benchmark was run on several experimental systems having small number of cores but including PGAS support in hardware like MVS-Express system [4]. Obtained result for this system is 5.5 times acceleration on 6 nodes.

The remainder of this paper is organized as follows. Section 2 reviews the details of PGAS-extension of OpenMP, Section 3 explains details of PGAS modified version of NPB UA benchmark, Section 4 provides results, conclusions and future work.

2 PGAS-extension of OpenMP and its implementation on Blue Gene/P

The proposed PGAS extension is a one-sided communication library that closely resembles to small subset of Cray SHMEM interface developed for Cray T3D

machine in 1991. Therefore we will describe our PGAS extension to OpenMP in terms of SHMEM functions. Actually we see that SHMEM provides the PGAS paradigm in the simplest way, giving user the illusion of global shared address space. The concept of local portion of that illusion shared memory is very important in PGAS programming concept. Each node can access its local PGAS portion of memory locally with read and write operations. Remote accesses are not so cheap as local that is why they should be done through special functions and cannot be done through ordinary machine pointers like in UPC language. When remote value is being read or written programmer should explicitly give the node number and offset of data for that node. That means that programmer should manually distribute data and should clear differentiate local accesses from remote ones, trying to minimize quantity of the latter.

PGAS extension should provide functions to execute the remote memory accesses like put and get. Also some synchronization primitives should be included the most important of which is barrier. Barrier not only provides synchronization but also guarantees that all put operations issued by other nodes with destination of current node can be read by current node. Standard property of delivering all put operations to remote nodes is guaranteed therefore only after executing barrier twice. Most important used functions are `shmem_put`, `shmem_get`, `shmem_barrier_all`. Initializing functions like `shmem_init`, `shmem_finalize` and `shmem_alloc` are provided.

Native implementation of SHMEM on Blue Gene/P system was not available, so it was implemented by authors on top of the DCMF communication layer. DCMF provides natively `DCMF_Put`, `DCMF_Get` functions. The message rate achieved by `shmem_put` with 8 byte data chunk equals 1 million puts per second for one core per node, and 2 millions per second for 2 and 4 cores per node. This numbers are relative small because of fact that all sends are done through DMA engine what is rather inefficient for small messages. Authors look forward to improve these numbers, particularly using lower level SPI layer instead of DCMF.

PGAS extension calls can be used with OpenMP directives independently, supposing SHMEM calls to be used to communicate among nodes and OpenMP to parallelize computations among cores of one node.

Unfortunately the original OMP version was relatively slow on one node of BG/P system giving only 1.22 times speedup on 4 cores of one node. That's why hybrid PGAS/OpenMP version gave only 10 percent performance gain when 4 cores were used. On 128 nodes PGAS version with 4 cores per node gives 2.26 acceleration relative 1 core per node mode. Because of that hybrid performance data is not provided in Section 4.

3 PGAS/OpenMP-version of NPB-UA

One time step of NPB UA benchmark consists of following parts: 1) advancing convection term using Runge-Kutta implicit integration 2) advancing diffusion term of equation using CG iterative solver 3) mesh adaptation on every 5th step. The most time consuming part is advancing of diffusion equation term, which includes solving linear equations system with a preconditioned CG iterative solver. For example one core of PowerPC 450 processor used in IBM Blue Gene/P supercomputer, have the

following time percentage of different subtasks on a sequential NASA version for a class C problem:

Mesh Adaptation	0.4 percent
Advancing convection term	24.4 percent
Miscellaneous computations	3.8 percent
Scatter and gather ops in main loop	1.9 percent
Advancing diffusion term	69.5 percent

Benchmark uses MEM (Mortar Elements Method) described in [3] and morton numeration of elements restored after each mesh adaptation. That helps in making application parallel. MEM is a nonoverlapping domain decomposition technique that couples different variational approximations in different subdomains. In this method MEM greatly increases the portability of SEM. In benchmark each element is covered by a Cartesian product of $(N+1)^3$ points. Points on edges of neighboring elements could differ because of using irregular mesh. Although most computations done with the element mesh points are data-parallel. The main concern of the MEM method is transfer of temperature values (advanced convection and diffusion term) between edge points of neighboring elements. To transfer these values two set of points are used: collocation points and mesh points. Collocation points consists of all mesh points of all elements, therefore their number equals to $N_{elt} * (N+1)^3$, where N_{elt} is a number of elements.

Obviously some border points of neighbor elements can coincide, but from method's point of view that will be different collocation points – temperature values is computed separately and the results will be different. Second set of points is called mesh and is actually a subset of collocation points. Mesh points are geometrically different points in space, unlike collocation points. Clear that subset of collocation points interior to all elements is equal to that subset of mesh points. Therefore the difference between two set of points are the border points, which are called mortar points, which are used a lot in MEM method.

At first time step mesh consists of only one element equal to a domain cube. Then at time step 0 and every 5th step mesh adaptation is performed. Sometimes mesh is refined when elements have high temperature difference, in other case mesh can be coarsened. One cubical element can be refined to eight smaller cubical elements and vice versa eight smaller elements can be transformed into one big element. The only condition is that neighbor elements should have factor of sizes not greater than two. Such adaptation provides us with a nonconforming mesh having finer resolution close to source of heat, while in other areas number of elements is relatively low increasing computational speed. All arrays in original benchmark code on Fortran can be divided to the following classes:

- arrays containing mesh elements and points information, mostly integer
- double precision arrays indexed with collocation points (tx-arrays)
- double precision arrays indexed with mortar points (tmor-arrays)

From programmer point of view almost all computational loops are indexed by collocation points or mortar points, and the loop iterations do not have any data dependencies, while each iteration works with data of its own element. Because of that parallelizing these loops is relative easy, we only need to distribute different elements among nodes. The only difficult operation which remains to be made parallel are so-called gather and scatter operations. These operations are the most essential for MEM

method and are used to map temperature values from one element to its neighbors and are performed on each CG iteration becoming one of the most time consuming operations in benchmark.

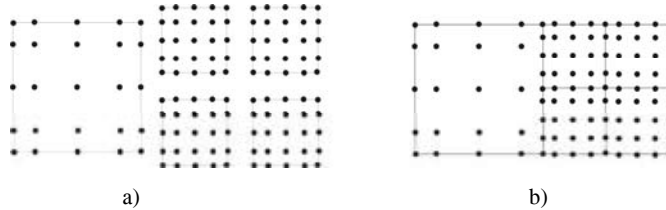


Fig. 1. Collocation points (1a) and mesh points (1b) on element faces

For the initial version it was chosen to leave the mesh adaptation sequential, to make effort on parallelizing scatter and gather operations as well as other computations. Spending 0.4 percent total time adaptation was not the critical part of the application, although thus our version was limited to 250x acceleration. Also that decision increases total memory consumption as all mesh points arrays should be located on each node and an additional all2all communication will slow down the parallel version. This all2all exchange is needed to collect calculated temperature values from every element on all nodes, because it is required by adaptation procedure, which should recalculate temperature values on new elements of the mesh. Despite of these limitations we expect benchmark to have rather good scaling.

Generally the code of gather operation could be described is follows:

```

do i=1,nelt
  do j=1,125
    tx(i,j)=a(j,1)*tmor(idmo(i,f(j,1)))
  &      +a(j,2)+tmor(idmo(i,g(j,1)))+...
  end do
end do

```

Where $nelt$ – is a number of mesh elements, $125=(N+1)^3$ – a number of collocation points in each element, tx array as index has number of collocation point, arrays a , f and g are constant, $idmo$ – index vector, indexed with collocation points, $tmor$ is indexed with mortar points. This loop as well as similar loop in scatter operation contains the most irregular and dynamic part of data accesses of the benchmark. The reason is usage of index vector $idmo$, which refers to mortar points in rather random order. Fig.2 that contains corresponding accesses from mesh elements to mortar points changing every 5th step as mesh is being adapted. This figure shows accesses for relative class W (less then 600 elements, less then 30000 mortar points). From the figure we could see the irregularity and how the data access pattern changes. Also we can find that accesses are not fully random as it is in Randomaccess benchmark. Many points are located on upper diagonal line or lower almost horizontal line. Studying the details of mortar point's numeration we can find out that lower line corresponds to mortar points located in element vertices as vertices are numbered first. And the diagonal line corresponds to face points of the elements. All other randomly located points on the figure correspond to face points of previously numbered neighbor elements. Provided with this analysis we can try to choose the best data distribution among nodes which not share their memory.

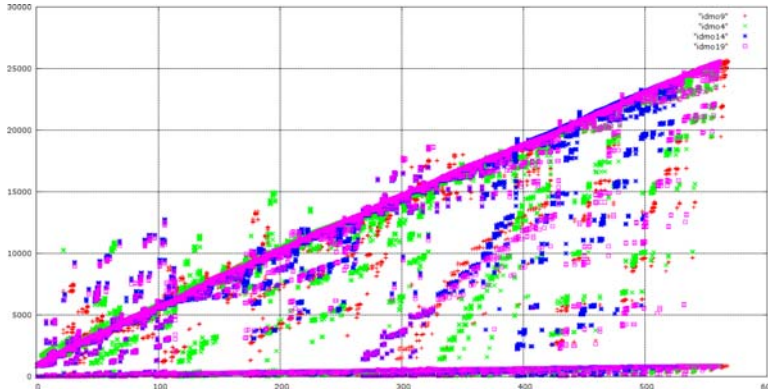


Fig. 2 Correspondence of mesh elements (Ox axis) to mortar points (Oy axis) in Gather/Scatter operations of NPB UA benchmark (class W) for different time steps.

Data distribution should aim the following: maintaining maximum number of local data accesses and reducing number of remote memory accesses (RMA), maintaining balance of remote memory accesses among nodes not to wait the last node on the barrier. It is very important to have equal number not only of outgoing remote accesses from all nodes, but also the number incoming accesses should be balanced among nodes.

Taking this into account following decisions were suggested: mesh elements and corresponding data arrays will have block distribution among nodes. Therefore access to this data indexed by collocation points in all loops will be local and sequential as in sequential version. That will happen because all loop interactions will be distributed among nodes in same blocks. It concerns the following arrays from gather pseudocode: tx, idmo. All non-local accesses will happen to tmor array which is indexed by mortar points in the gather loop provided. Indexes of such accesses are stored in big idmo index-vector, moreover all such accesses are done through this index-vector. We consider three distributions of mortar points and therefore mor-arrays among nodes: block, cyclic and «twiceblock».

Block distribution of mortar points is similar to elements distribution and is the simplest case. The benefits of such case besides its simplicity is providing locality for accesses found around diagonal line on Fig.2. Thus the number of RMA is significantly reduced. The major drawback of block distribution is unbalanced number of RMAs going to different nodes. Aforesaid vertices are numbered first and all accesses to them (lower line on Fig.2) get in several nodes with small indexes.

Cyclic distribution is organized as follows. The number of node storing mortar point is determined by taking remainder of mortar point index divided by total node number. This distribution lacks drawback of unbalanced RMA, but it doesn't take into account locality of accesses and most of them are non-local, increasing total network traffic and therefore communication time.

«Twiceblock» distribution tries to take in account peculiarities of mortar points numeration taking in account that vertices is numbered first. It is organized as follows. Mortar points being vertices (and having indexes 1..nVertex) and all other mortar points (indexes nVertex+1..nMor) have separate block distribution. Each node stores two blocks of points locally: one block (number i) of vertices and one (number

numNodes-i) for non-vertices. This distribution minimizes total count of RMA and has balanced number of RMAs from and to different nodes. Experiments showed that this distribution was best almost in all scenarios, so it was used further in all runs.

The program was modified so that any other custom distribution can be used. To do it several functions should be defined by user: islocalmor — given by a mortar index should say if this point is local to local node, pemor and localmor which do return node and local index on that node of a given mortar point by a global mortar index. Worth mention that every 5 steps distribution will be changed after mesh adapting, as total number of elements and mortar points will be changed but that makes no problem, as mortar points temperature is not used between time steps.

Aforesaid each node stores in it's local memory some part of tx array and corresponding part of idmo index-vector which controls all irregular accesses to tmor array. The proposed idea is to recalculate index-vector to separate local and remote accesses. In realization of Decoupled Access Execute DAE principle we want first to fetch all remote data needed to local memory, and after that to proceed the main loop. To do this each time after mesh adaptation, and creating new idmo index-vector we proceed with some action devoted to find all RMA to be fetched and save their indexes in a special sorted array containing only unique remote indexes. That array we do call access map. One of the benefit of fetching remote elements through access map is reducing communications, as often one node requires the same remote point in several (4-6) loop iterations. Therefore with an external fetch loop such elements will be fetched only once instead of being fetched 4-6 times. Obviously creating access map is an overhead, but it scales as number of nodes increase, and we should mention that access map is created only once after mesh adaptation while is being used several tenth times in gather operations. Access map is created by separate C++ code and is called from Fortran routines. Aside of creating the access map, another index-vector is created which stores indexes for local accesses and virtual indexes for remote accesses to special arrays in which elements will be fetched (Fig.3).

Access map is used in the beginning of every gather operation: with an shmем_get into special array morloc temperature of mortar points being fetched from remote nodes. After barrier all shmем_get operations will complete and the main loop is executed and it does not contain any remote accesses therefore do not suffer from long latency operations. To achieve this main loop was slightly modified and all accesses to tmor array via idmo index-vector were replaced with accesses through recalculated xidmo vector as in Fig.3. The modified code of gather operation looks as following:

```

        call shmем_barrier_all
c.....request outstanding tmor accesses to morloc array
        do i = 1,mormap_size
            pe = pemor(mormap(i))
            call
shmем_double_g(morloc(i),tmor(localmor(mormap(i))),pe)
        end do
c.....gets synchronization
        call shmем_sync_gets
        do ie=startelt,endelt
            do j=1,125
                tx(i,j) = a(j,1)*readmor(xidmo(i,f(j,1)))
            
```

```

&                                +a(j,2)+readmor(xidmo(i,g(j,1)))+ ...
    end do
end do

```

Where readmor function reads either morloc array either local tmor elements.

Another advanced trick that was used is replacing shmем_get operations with shmем_put operation. To get this done the inverse access map is being calculated after each mesh adaptation. This reduces communication time as get requires two network packets while put requires only one.

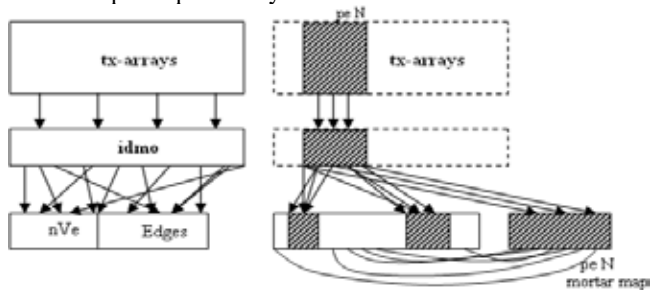


Fig. 1 – Index-vector accesses in sequential version (a) and local and remote accesses through recalculated index-vector (b) in PGAS version of benchmark

Scatter is inverse operation to gather and uses the same idmo index-vector. The only complication is that instead of reads we do need to make a double precision addition to the remote array element. In OpenMP version for synchronization OMP_ATOMIC directive is being used. As no atomic floating point operations were supported in hardware we proceed scatter with the same trick with recalculated index-vector. All double increments are done to special local array morloc and then all these elements are being transferred once to the remote nodes. After global barrier nodes to add received elements to its own local tmor elements. Later we will consider using DCMF_Send operations to implement such atomic operations, but for current version DCMF_Put is being used aside with a lot of memory consumed for temporary data transfer.

4 Results and Future Work

Fig. 4 illustrates absolute performance in millions of operations from number of cores used for BG/P system with proposed PGAS version, run in three modes: 1, 2 or 4 cores per node used. On BG/P system partitions from 128 nodes to 1024 were available to author. Lower number of nodes it was unable to run due to low number of IO-nodes on BG/P system. Class C problem was run having following parameters: element count less than 33500, 4 millions collocation points total, mortar points count 1.26 millions, maximum element refinement level 8. Memory consumed by sequential version 508 MB, PGAS version consumes 166 MB mesh information + 150 MB PGAS area on each node mostly for storing array to emulate atomic operations with floating point. For comparison performance of SGI Altix 3700BX2b with original OpenMP version is provided. That data was provided by from NASA NAS Advanced

Supercomputing Division. Maximum performance on Altix system achieves 207 millions operations per second and is achieved on 256 Itanium2 cores.

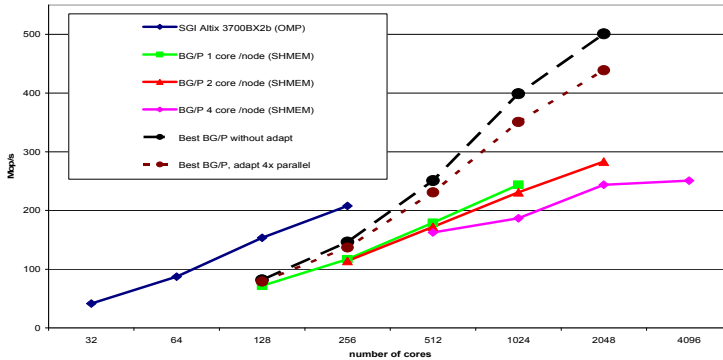


Fig.4 Absolute performance of NPB-PGAS UA (class C) on Blue Gene/P and NPB-OMP on SGI Altix 3700bx2.

From the figure we can see that SMP and DUAL mode are almost identical by performance with equal number of cores used. VN mode is slower, showing contention of four cores for memory and DMA engine of the network router.

Maximum performance of PGAS version is 283 millions operations per second and is achieved on 1024 BG/P nodes run in DUAL mode. With 4096 cores used contention gives a reasonable slowdown. Also class C comes to limit of its scaling as every core processes less than 8 elements and 315 mortar points, having to small granularity even for a Blue Gene/P system.

Given sequentially performed mesh adaptation, it also limits the scaling of the benchmark on large number of cores. For example on 128 cores adaptation takes only 14 seconds out of total 122, and for 2048 cores adaptation takes 14 seconds out of 31 total. On 1024 cores absolute scaling achieves 100 times, excluding adaptation time we do have 171 times faster relative to sequential version. Planned levels of performance are provided on figure 4 in case of zero time adaptation and in case it can be parallelized among 4 cores of node. That shows that level of 400-500 millions operations per second can be achievable with class C on 1024 nodes of BG/P, what means only one rack is used.

It was demonstrated that PGAS-extension of OpenMP can be highly productive and efficient for applications with irregular data accesses. On 2048 BG/P cores 37 percent faster result was achieved that was previously known on 256 Altix cores. On future hardware with PGAS supported in hardware results can be even higher.

Future work will include but not limited to: parallelizing mesh adaptation at least among cores of one node, reducing memory consumption, implementing floating point atomics via DCMF_Send, making visualization of data distribution using system described in [5], implementing message aggregation, exploiting future architectures from Cray if the access will be available.

References

1. Kumar, S., et al, The deep computing messaging framework: generalized scalable message passing on the blue gene/P supercomputer. In Proceedings of the 22nd Annual international Conference on Supercomputing. ICS '08. ACM, New York, NY, 94-103.
2. Unstructured adaptive (UA) NAS Parallel Benchmark, Version 1.0 : technical report : NAS-04-006 / NASA Ames Research Center, Moffett Field ; H. Feng, R.F. Van der Wijngaart, R. Biswas [at al.]. – CA, 2004. – 17 p.
3. Feng, H., Mavriplis, C., Feng, R., and Biswas, R. 2006. Parallel 3D Mortar Element Method for Adaptive Nonconforming Meshes. J. Sci. Comput. 27, 1-3 (Jun. 2006), 231-243.
4. http://www.kiam.ru/MVS/research/mvs_express.html
5. Dzhosan O.V., Popova N.N., Korzh A.A. Hierarchical Visualization System for High Performance Computing // pros. conf. ParCo 2009, France, Lyon, 2009

From Process Flow Diagrams to BPMN

Vladimir Dimitrov¹,

¹ University of Sofia, Faculty of Mathematics and Informatics, James Bourchier Blvd. 5,
1164 Sofia, Bulgaria
cht@fmi.uni-sofia.bg

Abstract. Process flow diagrams are used by DG TAXUD for specification of business processes. For specification of business processes, software development tools use BPMN. Conversation of process flow diagrams to BPMN diagrams is vital to be done in efficient and flexible way. This paper present such an approach based on an example.

Keywords: formal specification, BPMN, process flow diagrams.

1 Introduction

European Commission's Directorate-General for Taxation and the Customs Union for description of models of Business process model view uses the next notations:

1. Process Flow Diagram, which is providing a graphical representation of the Transit business thread. It shows how the business arranges its processes to respond to external events and to produce results. This specific technique allows for full understanding of the Transit business before detailing the information technology solution supporting those processes. Many businesses have similar processes, but the arrangement of the processes—the dynamics or process flow—may be very different.
2. Textual description of the components of the Transit business thread and shown on the process flow diagram:
 - the (major and minor) events that launch the thread;
 - the processes that are involved in the thread;
 - the (major and minor) results produced by the thread.
3. When applicable, specific assumptions, constraints or remarks will be provided for each of the elements.

The Data model view will be limited to the following two models:

1. The model of the State Transition Diagram presents the life cycle of the data, showing the different states and the way it goes from one state to the next one. State Transition Diagrams are provided for the most significant process threads. In practice, State Transition Diagrams are not used.
2. During the execution of the Transit business threads, information is exchanged between processes. The Structure of the information to be exchanged together with its symbolism is not provided here – it is not subject of this paper.



The event is represented as a large arrow pointing from left to right and is drawn in the column of the location(s) where it happens.

Each event name starts with an E_ followed by the name.

An event that can happen at different locations is represented on the diagram as one event overlapping the different location columns where it can happen, e.g. the event “E_Transit consignment arrives at destination” can happen at the ‘Premises of the Trader at Destination’ or at the ‘Actual Office of Destination’.

An *elementary business process (EBP)* [ⓑ], also called *process* in the document, is represented as rectangle containing its identification and its name.

Each EBP is identified by a string of the format ‘xxttssss’ where:

- ‘xx’ identifies the business area, which can be: CO (Core business); GU (Guarantee management); CS (Central services); SA (System administration).
- ‘tt’ identifies the process thread in that business area (e.g. CO02 - “Process arrival”);
- ‘ssss’ identifies the EBP in the process thread (e.g. CO020102 - “Process arrival notification”).

The EBP name is composed by a verb followed by a complement (e.g.: Process arrival notification).

A process that can be performed at different locations is represented on the diagram as one process overlapping the different location columns where it can be processed, e.g. the process CO1A1001 - “Verify goods and/or supporting documents” can be performed either at the ‘Premises of the Trader at Destination’ or at the ‘Office of Departure’.

A *result* [ⓒ] is a business product put at the disposal of other processes, e.g. “R_Presentation rejected at destination”.

The result is represented as a large arrow pointing from right to left and is drawn in the column of the location(s) where the result will be used.

The result name starts with R_ followed by the name.

A result that can be used at different locations is represented on the diagram as one result overlapping the different location columns where it can be used, e.g. the result “R_Presentation rejected at destination” can be used at the ‘Premises of the Trader at Destination’ or at the ‘Actual Office of Destination’.

A *process flow break* [ⓓ] occurs when one process in the flow is complete, but the next process cannot start before an event occurs to restart the process flow.

When the time interval allowed for the break has elapsed, an internal event is created (in the given example ‘E_Document Timer expired’) and used as a trigger for the next process. If some condition occurs during the time interval that allows no longer waiting for its end, another internal event is created (in the given example ‘E_Document received’) and used as a trigger for the next process.

The process flow break is represented as a large “U” on its side. It is followed by the events that will cause the process flow to restart.

A *flow connector* [ⓔ] is used to clarify the diagram when the next component in the process flow is too far from the previous one. It must be related to process belonging to the same business process thread.

The flow connector is represented as a small house on its roof. It is labeled with a number and both components (the previous one and the next one) are in the same diagram, maybe on different pages.

An *iteration* (F) describes the case where sequences of processes need to be repeated a number of times, as if in a loop.

The loop may continue a predictable number of times (in this case the statement specifying the number of iterations will start with the word 'For'), or it may continue an unpredictable number of times until some expected verifiable condition is satisfied (starting with the word 'Until').

The process flow diagrams indicate also in background the locations where the processes are executed. Processes are put in the column, or across the columns, whose title identifies the *location* (G) where it is executed.

The above diagram shows that the EBP CO020102 "Process arrival notification" is performed at the location of the "Actual Office of Destination". This process includes, when the arrival is accepted the sending of an information (IE06) to the location of the "Office of Departure". Upon reception of that information, the EBP CO020701 "Store and forward arrival advice" is performed at the location of the "Office of Departure".

The flows are represented using a line terminated with an arrowhead which starts from one component and leads to another component of the process flow.

1. A *mandatory flow within the same location* (1) is represented by a narrow solid line; in this context, 'mandatory' means that the succession of the component is imposed (in the example, the EBP CO020701 "Store and forward arrival advice" always produces the result 'R_Office of Departure notified of arrival (or diversion)').
2. A *mandatory flow between different locations* (2) is represented by a wide solid line. This flow between locations is always labeled with an identifier of the information to be exchanged in the form IE_{xx}.
3. An *optional flow within the same location* (3) is represented by a narrow dashed line; in this context, 'optional' means that the succession of the component depends on a certain condition. So, this flow is always labeled with the name of the condition.
4. An *optional flow between different locations* (4) is represented by a wide dashed line and is always labeled with the name of the condition (because the flow is optional) and the identifier of the Information to be Exchanged.

It must be noted that, depending on the circumstances encountered by an EBP, the latter can produce different sets of flows. As explained above, each optional flow is labeled with the name of a condition. When two or more flows leaving one process are labeled with exactly the same name of condition, which means that they are produced simultaneously (see condition "Arrival accepted" (A)).

An *Information to be Exchanged (IE)* represents the flow of information between two EBP's taking place in different locations. No assumption is made on the communication medium used to carry out this flow of information. The content and structure of all IE's are detailed in Appendix B 'Logical Data Model / Functional Structure of Information To Be Exchanged'. The following remarks apply to all IE's:

1. all IE's are supposed to comply with the structure, the conditions and the rules described in Logical Data Model / Functional Structure of Information To Be Exchanged. This compliance is always checked on the reception side of the IE. In case of non-compliance, the IE is rejected, an advice of non-acknowledgement (NACK) is returned to the sender, the process flow is interrupted, and exception handling measures must be taken;
2. an IE may contain some free text written in the language of the originator.

3 Conversion

In the next following, we will try to converse step by step the diagram given at Fig. 1, and then we will comment how to convert some additional elements, that are not used in this diagram. IBM WebSphere Business Modeler Advanced Version 7.0 is used for modeling purposes. It supports BPMN notation and example conversion is developed with it – see Fig. 2.

A Process flow diagram is converted to BPMN diagram. Process flow swimlanes are modeled as swimlanes in BPMN. In the example, swimlanes are locations, in BPMN it is recommended swimlanes to be note workers, but it is acceptable to be used for locations.

The process starts with the event “E_Transit consignment arrives at destination”. This event could happen at two places: Premises of the Trader at Destination and Actual Office of Destination. This means that some software at these places generate the message IE07 that have to be received and delivered to process CO020102. A receive task is put at Premises of the Trader at Destination and another at Actual Office of Destination. In such a way, it is supposed that on both locations there are client that receive from the outside information needed to generate and deliver IE07. This does not mean that receive tasks receive IE07, but that they deliver this message.

Another way for presenting these events is through an input entry on the diagram boarder, but this means that IE07 is received from the process environment. Usually this approach is used later in more detailed modeling phase when system boundaries are clarified.

As functionality, IE07 could be generated on both above mentioned locations, i.e. the Trader has two options for starting the process. Independently of the place IE07 is delivered to the process CO020102. This is presented with merge gateway.

These are options for modeling Process flow events and especially when they are placed on swimlane boarders. One more comment is that there are two options, later, when these events start a new copy of designed process or simply need to correlate finding the right copy of the process.

Every elementary business process is presented in BPMN as a task. This task could be converted later to global process, global task, global service, local process, local human task, local business rules task or compensation activity. It depends of its nature or from design decisions taken later on. For example, if the task contains human-computer interactions it can be represented as human task. Sometimes it is impossible when the task is huge one and contains sophisticated business logic, but not only

human-computer interactions. In the last case, some other option, like process or service, has to be used.

If the presentation is invalid CO020102 has to send IE08. This means that the message has to be delivered to the Trader. Usually, this is the location from where IE07 has come. This could be modeled with a simple decision gateway or with two outputs to both destinations, which means that the decision where to send IE08 is taken internally in CO020102. On the BPMN diagram, IE08 is send to both locations with the idea that every message has to be delivered to Trader's system. When IE08 is received at Actual Office of Destination, the result is to stop the process, but when it is received at Premises of the Trader at Destination, this does not means that Trader's system has to be stopped. That is why in the first case Terminate event is used, and in the second one – End event.

That is the way how Process flow results can be modeled. Results are modeled as tasks, because, frequently, some clearing is needed, when a process stop or terminate.

If arrival is accepted, then IE06 has to be sent to CO020701. Here CO020102 does not generate all its outputs, it generates different outputs depending of its inputs and processing.

After CO020701, we have to model the cycle. It is to some extend cumbersome. First, a repository for IE24 is created from where the process in the cycle can access it. The process CO020702 is put in the cycle activity, control flow is given to the cycle, and then CO020702 is executed for each declared country of transit, how it is specified in the loop condition.

Another possibility is IE07 to carry some diversion; this means that no new instance of the process has to be created, instead of that IE07 message has to correlate with its corresponding instance of the process. In this case, CO020102 generate IE02 and deliver it to CO020300 for further processing. The last one after some processing delivers IE06 to CO020701. In such a way, CO020701 accepts two different inputs that have to be processed differently.

The last flow is from CO010201 to the timer. When arrival is accepted the timer is started to wait for the documents. If the timer expires, then CO022001 is executed to ask for documents. In both cases, control flow is back to CO010201. This will be implemented with some difficulties in BPEL, because a cycle is created, but in this mode (Advanced Modeling) IBM WebSphere Business Modeler Advanced does not issue an error.

Timer events are not presented as some elements, because they are the alternative outputs from the timer, only some comments are put on the destination tasks.

All messages that have states are modeled with states and this is put in the BPMN diagram.

Finally, some of the flows are obligatory and other ones are optional. This could not be modeled.

4 Conclusion

In this informal presentation, we have shown that Process flow diagrams with some modeling can be converted to BPMN diagrams. The proposed kind of conversion is a designer vision and highly depends of the concrete implementation.

To some extent, it could be said that Process flow diagrams are at higher level of abstraction than BPMN ones. Process flow diagrams are more compressed and they are applicable to specify broader policies at EU level than concrete national ones. But in any case, real implementation of processes usually start with BPMN specification, that is why we need of an iteration from Process flow diagrams to BPNM diagrams to specify implementation of EU policies at national level.

Acknowledgments. This research is supported by Project 240/2010 “Development of Grid infrastructure for research and education” funded by the Scientific Research Fund of University of Sofia.

References

1. Taxation and Customs Union DG, FITSDEV Project. Specifications, Development, Maintenance and Support for communication and information exchange systems in the taxation and excise area, <http://www.uni-mannheim.de/edz/pdf/sek/2007/sek-2007-0696-en.pdf>
2. BPMN, www.bpmn.org
3. IBM WebSphere Business Modeler Advanced Version 7, <http://www-01.ibm.com/software/integration/wbimodeler/advanced/features/>

Load Balanced Resource Management for Cloud Systems¹

Radko Zhelev^{*}, Vasil Georgiev^{**}

^{*}Institute on Parallel Processing, Bulgarian Academy of Sciences

^{**}University of Sofia “St Kliment Ohridski”

Abstract. In this paper we propose a distributed topology and load balance scheme for management of resources in Cloud systems. We combine the benefits of peer-to-peer clusters with a two-level hierarchical topology, as a best balance between scalability and efficiency for a resource management system inside the Cloud. We define a suitable load balance scheme for dynamic distribution of the work load in each management cluster. The load balance algorithm gracefully takes advantage from the context of the distributed work field, and works completely on calculation basis implying no overhead for data synchronization messages or I/O access to shared memory. This affects in hitting up the performance capabilities and the limits of scalability of our clusters. Additional benefits we introduce, is enabling of high performance non-replica caches for arbitrary functional modules handling resources inside our topology. We can free ourselves from the complexity of shared-memory caches, and be able to use non-replica caches relying on the load balance algorithm to lead us to the exact place where data is cached.

Keywords: Load Balance, Cloud Systems, Resource Management, Distribution Topology.

1. Related Work Overview

1.1 Resource Management Topologies

Existing taxonomies of distributed systems [2] [3], initially classify the organization of resources into **centralized** and **decentralized**. Centralized management systems (as well as centralized clusters (Fig. 1a)) have one single machine functioning as a central manager that handles the whole synchronization and work distribution at

¹ This paper is supported by the project 156/2010 of the University of Sofia



central place. Centralized management is simple and easy for realization, but suffers from lack of scalability due to the load capacity limitation of the central manager and can not ensure reliability and high availability due to the single point of failure [4]. A taxonomy of Resource Management Systems [1] separates the decentralized organization of resources into **flat**, **cells** and **hierarchical**. In a flat organization all machines can directly communicate with each other without going through an intermediary. This organization we can refer as a peer-to-peer cluster (Fig. 1b). In a cell structure the machines within the cell communicate internally using flat organization, but there are designated machines acting as boundary elements that are responsible for all communication outside the cell. We can refer such structures as mediated clusters (Fig.1c). Mediated clusters usually scale better than peer-to-peer clusters because of the minimized I/O synchronization. But for this scalability they pay with lower performance because of the intermediate redirections from boundary nodes to internal nodes, and with lower availability since dropping of boundary elements causes loss of the whole cluster. In a hierarchical organization, machines in the same level can directly communicate with the machines directly above them or below them (Fig.1d). Most current Grid systems use this organization since it has proven scalability. Of course as much the hierarchy levels are, as slower the performance becomes due to the increased number of redirections as requests travel forth and back.

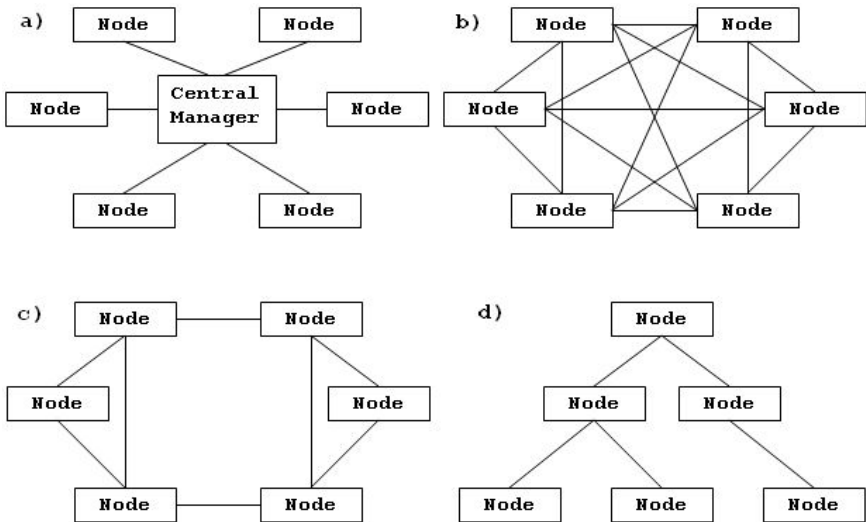


Fig. 1. a) Centralized Cluster, b) Peer-to-Peer Cluster, c) Mediated Cluster, d) Hierarchy

1.2 Load Balance

Load balancing is the process of distributing the workload of a system among the nodes (workers, processors) participating in the distributed topology. The purpose of

the load balancing is to minimize the execution time of requests to the system and to provide efficient utilization of the dedicated working nodes. Load balance strategies are generally classified upon several characteristics.

The two major categories for load-balancing algorithms are **static** and **dynamic** [5] [7]. Static load balancing algorithms define sharing of the load on preliminary basis, without runtime decisions. Dynamic load balancing considers the runtime state of the system and dynamically changes the distribution of work among worker nodes at runtime. As a result, dynamic load balancing algorithms can provide a significant improvement in performance over static algorithms. However, this comes at the additional cost of collecting and maintaining load information, so it is important to keep these overheads within reasonable limits.

Dynamic load balancers usually take into consideration several types of information to make distribution decisions. They monitor the availability of worker nodes and the load of each node. They also may consider the required capacity for the planned task and sometimes even the network connectivity bandwidth. For dynamic load balancing, there are three important parameters which usually define the strategy a specific load balancing algorithm will employ. **Centralized** versus **Distributed** is the first consideration to be taken into account [6]. In a centralized scheme, the load balancer is located on one master workstation node and all decisions are made there. This strategy is applicable for the centralized cluster topology (Fig. 1a). In a distributed scheme, the load balancer is replicated on all workstations. The next consideration for dynamic load balancers is whether to provide functionality for **rebalancing and migration**. This refers to migrating a work from one node to another [9]. Task migration could be done for balancing the load as well as for enabling failover (if a responsible node gets crashed). The third consideration for distributed load balance is based on the question - who makes the load balancing decision. Two policies are generally employed - **sender-initiated** and **receiver-initiated**. In sender-initiated policies, congested nodes attempt to move work to lightly-loaded nodes. In receiver-initiated policies, lightly-loaded nodes look for heavily-loaded nodes from which work may be received [8]. It is typical for sender-initiated policy to show better performance than the receiver-initiated at low to moderate system loads, and vice versa. The reason for this is in the reversed probability of finding a lightly-loaded or heavily-loaded node in the different cases.

Which strategy to choose is a matter of consideration upon the particular system use cases. Static load balance has no synchronization overhead, but can not provide redistribution and failover, and for most cases would not bring optimal workers utilization. Centralized dynamic distribution could be suitable for many systems, but they will be conceptually limited about scalability. Regarding migration functionalities, applications that exhibit lengthy parallel jobs usually benefit in the presence of a job migration system. However, applications with shorter tasks usually would not prefer the expense of job migration support. As always, different requirements are met optimally by different solutions.

2. Distributed Topology

Cloud systems typically consist of one or more intentionally built datacenters, physically spread in strategic geographical locations over the world. For instance one of the biggest Cloud owner companies – Amazon, has exactly three datacenters at the time of present. Cloud datacenters are usually built with the purpose of business servicing, and contain farms of hardware and equipment assembled internally in an optimized environment. These circumstances lead us to the idea to place one cluster of Management Servers in each datacenter to handle the management of resources there. The cluster should handle the organization of resources in the whole datacenter, sharing internally the load of management activities. Since we potentially have more than one datacenters, we would need additional layer to consolidate the results from all clusters into a centralized view to the Cloud system administrators. Figure 2 shows our distributed architecture:

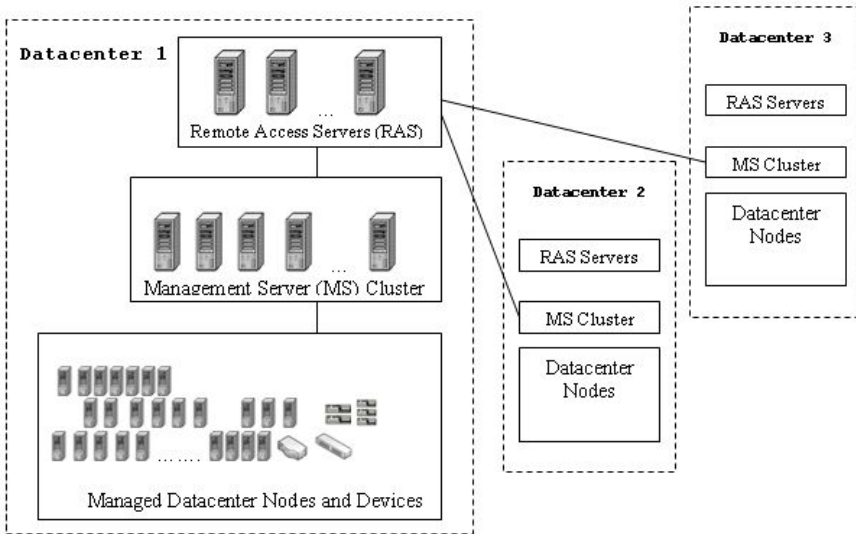


Fig. 2. Distributed Architecture

A Cloud system and its resource management system consist of the following component roles:

Managed Datacenter Nodes and Devices – these are the managed datacenter nodes along with all physical and logical resources available for monitoring and administration - hardware/virtual resources, service applications, tenant users sharing resources in isolation, service configurations, as well as completely logical resources like maintenance procedures.

Management Server (MS) Cluster – a cluster of Management Hosts (dedicated nodes) in the datacenter called Management Server that handles directly the management of datacenter resources. MS cluster is responsible for maintaining resource information in databases, synchronizing resource data with the real states of

the underlying resources, scheduling management operations, capacity estimation for the end-users servicing, performing resource filtering and so on. We choose an organization of **peer-to-peer** cluster with the purpose of sharing internally the load of resource management activities, and providing efficient utilization and failover.

Remote Access Servers (RAS) – they provide system interface to the Cloud administrators for accessing the management and monitoring functionality – via GUI, Web Browser or arbitrary administration tools. RAS Servers are capable to perform inter-datacenter connection and consolidate the results from all MS Clusters into a centralized view.

3. Load Balance Algorithm

In this section, we will focus on the load balance algorithm in our Management Server cluster. As already mentioned we employ a peer-to-peer cluster organization, placing one such cluster of dedicated management server hosts in each datacenter, with the responsibility to handle all physical and logical resource subject of monitoring and administration. The size of the cluster (in terms of number of participating hosts) may vary with accordance to the volume of the managed datacenter and the variety and amount of resources that are subject of administration.

3.1 The Cluster State

In the related work overview section, we noted that dynamic load balance algorithms usually monitor the availability and the current load of each node. One of the major cornerstones in our algorithm is that we will skip the monitoring of the work load and will only check our hosts for availability. In one hand, we will show that we still achieve completely efficient utilization of the cluster, and in the other - we will naturally hit up the cluster performance and scalability by removing the burden of monitoring load of its hosts.

Our cluster state is defined as follows. Considering we have a cluster of N hosts, we can always order them in a sorted sequence assigning a static index to each cluster host – $\{C_0, C_1, \dots, C_N\}$. Ordering could be done by some arbitrary parameter – ip address, system id, etc.

We define two possible states of a cluster host: 1 – means host is available, 0 – host is not available (down or not operable). A **Cluster State** we will call the sequence of Booleans showing the availability of each host at the respective index. For instance sequence of $[1, 1, \dots, 1]$ will mean that all hosts are up and operable, while sequence of $[0, 1, \dots, 1]$ would mean the host and index 0 is not available. The cluster hosts are connected in a peer-to-peer network exchanging periodic ping-pong messages to each other. We can consider that every host in the cluster ‘knows’ the entire cluster state, since everyone can detect that some other host gets down - when response to the ping is not received in a predefined time frame.

3.2 The Context of a Resource-handling Environment

Resource Management Systems in distributed computing environments usually use resource abstractions for representing administered resources and for manipulating them in a uniform way [1]. Resources are usually represented by a set of properties describing the resource state (schema-based) and optionally they may have set of operations available upon the resource (object-based). What is common for all types of resource representations is that resources always have some identification – a namespace, a global access URI, or simply an ID.

3.3 The Algorithm

Concept and requirements

The concept is based on responsibilities. We define that every cluster host should be responsible for a certain subset of the resources at any moment in time. To be efficient, our balancing algorithm should ensure **even distribution**, i.e. equal subset of resources handled by every cluster host. In the analysis section we will prove that even subsets of resources assigned to each cluster hosts, reflects in even workload distribution of the cluster hosts. For the purpose of **determination**, we also need to ensure unambiguous subset of resources assigned to a given cluster host, as well as unambiguous responsible host for every given resource. This strategy also brings an important characteristic of our distributed system – we have a **single place of responsibility** (which is not single point of failure) for every given resource. This characteristic prepares the ground for the further performance advantage of using non-replica caches. And finally, our algorithm must provide an efficient failover. This means that when a host from the cluster goes down or gets blocked in some thread deadlocks, or sticks in an endless cycle (thus becoming not operable and able to respond to the pings of others), we need a mechanisms to reassign the subset of resources from the dead host to the other operable hosts. Rebalancing should still keep the even distribution, and in order to be efficient it must be done with **minimal failover redirections**. We don't need an algorithm that globally swaps all resources between the clusters hosts, causing a total chaos in our cluster when some host goes down. Instead, we need an algorithm that redirects only the 'orphaned' resources, not causing impact upon the ones that are stably handled by the working cluster hosts.

Description

We build all permutations of cluster host indexes and order them lexicographically [11], assigning to each permutation an index number. For N in count cluster hosts, we would have N! in count permutations. Here us their lexicographical order:

$$P_0 = C_0, C_1, \dots, C_{n-2}, C_{n-1}. \quad (1)$$

$$P_1 = C_0, C_1, \dots, C_{n-1}, C_{n-2}.$$

...

$$P_{n-1} = C_{n-1}, C_{n-2}, \dots, C_1, C_0.$$

Next step is to use some hash function that maps a Resource ID to an Integer hash code:

$$\text{hash}(\text{Resource ID}) \rightarrow \text{Integer} \quad (2)$$

Any hash function [12] that produces chaotically spread integers would do the job. The remainder produced by dividing the hash code to the number of permutations (n!) would give us a certain permutation index among the lexicographical order of permutations:

$$\text{hash}(\text{Resource ID}) \bmod n! \rightarrow R \in [0 - n!-1] \quad (3)$$

As a result, we have a compound function (p) that maps a Resource ID to a certain permutation of cluster hosts.

$$p(\text{Resource Id}) \rightarrow P_R = \{C_{R_0}, C_{R_1}, \dots, C_{R_n}\}, \text{ where:} \quad (4)$$

$$R = \text{hash}(\text{Resource Id}) \bmod n!$$

Interpretation

The mapping of Resource Id to a certain permutation of cluster hosts (formula 4), we interpret as static order of responsible hosts for a given resource (identified by Resource Id). In the mapping $p(\text{Resource Id}) \rightarrow P_R = \{C_{R_0}, C_{R_1}, \dots, C_{R_n}\}$, we nominate C_{R_0} as the Main responsible host for that resource. This means that only C_{R_0} (in case C_{R_0} is available) is capable to handle the management activities related to the resource 'Resource Id'. C_{R_1} is considered Secondary responsible host and handles the resource only if C_{R_0} is down. C_{R_2} is considered a Third responsible and takes the handle in case C_{R_0} and C_{R_1} are not available. And so on..

Resource handling

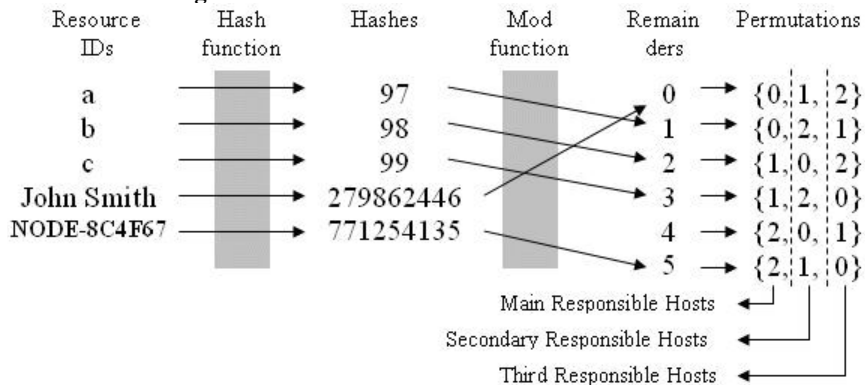


Fig. 3. Responsibility Distribution

Figure 3 illustrates determining the sequence of responsible hosts for resources in a three-hosts cluster. Resource IDs are initially passed through a hash function that produces respective integer Hash codes. Hash codes are then passed through the modulo function (mod 3!) that assigns a certain permutation number from the lexicographical order of permutations of all three hosts. The responsibility distribution ensures single and determined responsible host for the resources at any moment in time (i.e. for any state of the cluster).

Even Distribution

Main Responsible Host



P ₀ =	0	1	2	3
P ₁ =	0	1	3	2
P ₂ =	0	2	1	3
P ₃ =	0	2	3	1
P ₄ =	0	3	1	2
P ₅ =	0	3	2	1

P ₆ =	1	0	2	3
P ₇ =	1	0	3	2
P ₈ =	1	2	0	3
P ₉ =	1	2	3	0
P ₁₀ =	1	3	0	2
P ₁₁ =	1	3	2	0

P ₁₂ =	2	0	1	3
P ₁₃ =	2	0	3	1
P ₁₄ =	2	1	0	3
P ₁₅ =	2	1	3	0
P ₁₆ =	2	3	0	1
P ₁₇ =	2	3	1	0

P ₁₈ =	3	0	1	2
P ₁₉ =	3	0	2	1
P ₂₀ =	3	1	0	2
P ₂₁ =	3	1	2	0
P ₂₂ =	3	2	0	1
P ₂₃ =	3	2	1	0

Fig. 4. Even Distribution

Figure 4 shows the lexicographical order of permutations of four cluster host indexes. The first column in the tables is the main responsible host defined by each permutation. The nature of the permutations is such that we have equal number of permutations starting respectively with index 0, 1, 2 and 3. Adding to this the fact, that division of chaotic integers (resource hash codes) by 4 produces every remainder (0, 1, 2 or 3) with equal probability, we may say that we have equal subsets of resources assigned to their Main responsible hosts in the cluster.

Failover

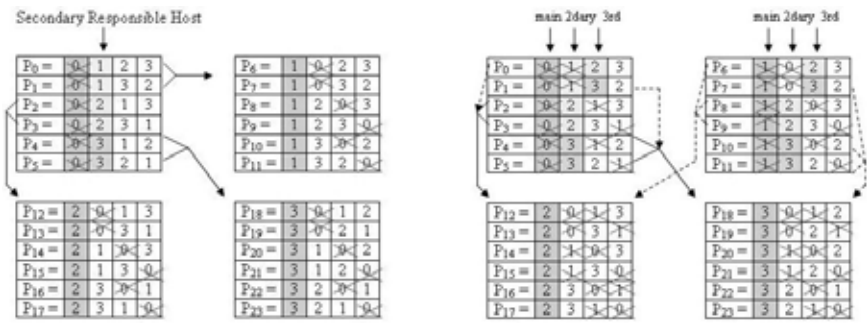


Fig. 5. Failover redirections: a) Host0 is down; b) Host0 and Host1 are down

Figure 5 illustrates the failover redirections when cluster hosts get down. Again the nature of the permutations ensures that our rebalanced distribution remains even, i.e. after rebalancing, cluster hosts continue to bring responsibilities upon equal subsets of resources. Another advantage of major importance is that this method achieves minimized number of redirections. Only resources handled by the crashed hosts get redirected without building a completely new responsibility scheme for the new cluster state.

4. Utilizing non-replica caches

Resource Management Systems ambitious to provide efficient management of resource data, and respectively – a competitive system performance, usually need to implement runtime caching of resource data in order to minimize the drawbacks from slow database operations. Implementing runtime caches in distributed systems is not an easy task and is related to maintaining the consistency of the cached data via synchronization messages or common access to a shared memory. One of the major advantages we introduce with our load balance algorithm is that we have a single-place responsibility for a given resource at any moment in time. This prepares the ground to free ourselves from the complexity of maintaining distributed caches, whose synchronization would require message exchanges, dramatically decreasing the system performance and making the situation worse and worse with the growing of our cluster. Instead, we can use non-replica caches relying on the load-balance algorithm to lead us to the exact place where data is cached. On every cluster host we are free to utilize some efficient non-distributed caching algorithm like Least Recently Used [10] to increase the “hit rate” of cached records. Here is a simple use case:

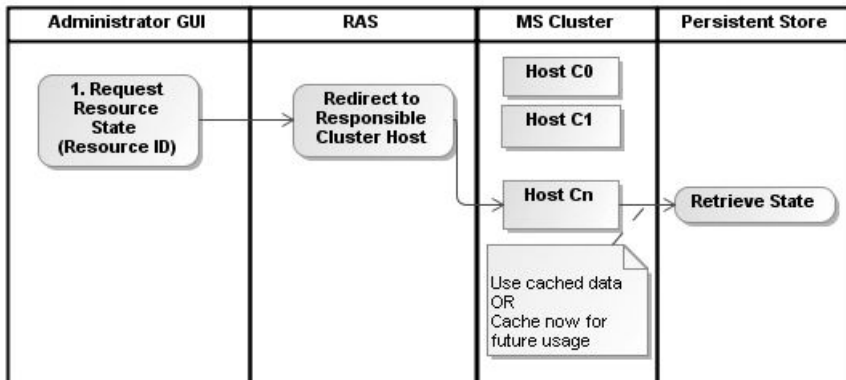


Fig. 6. Utilizing non-replica caches

In figure 6, the administration GUI operated by Cloud system administrator sends a request to the Remote Access Server (RAS) requesting the state of a given resource. The RAS server redirects the call to the proper host of the Management Server (MS) cluster – the one that is currently responsible for the requested resource regarding the cluster state. The MS host must either retrieve the state from a persistent storage or to access the information from the underlying resource directly. Here is the place where MS may use cached data instead of performing heavy I/O operations for retrieving the resource state. Synchronization of cached data is not needed since our load balance algorithm ensures that requests for that resource are directed exactly to the proper responsible host and this resource data is not accessed from other place. Of course when a cluster state changes (which is considered to be not a highly frequent operation), cached data could be cleaned to prevent inconsistencies after rebalancing accompanied by changing of the single place responsibilities.

5. Analysis

Our study is lead by the motivation to find a distributed and load balanced resource management architecture that is most suitable for Cloud systems. Our distributed topology is made with respect to the Cloud physical specifics – Clouds are leveraged on the top of intentionally built datacenters with internally optimized environment. We keep away from a centralized type of management due to the lack of scalability and the single point of failure problem. Although scalability is a proven characteristic of the hierarchies, arbitrarily leveled hierarchies impose latencies as the requests travel through the hierarchy levels. For this purpose, we choose a topology that is best balance between scalability and performance efficiency for the Cloud. We build a peer-to-peer cluster for managing the resources inside the datacenter, and combine it with a two-level hierarchy consisted of RAS servers on the first level and MS clusters on the second. Thus, we take advantage from flat and hierarchical topologies, employing them with respect to the Cloud specifics. We benefit from dynamic load balancing in the flat cluster inside the optimized datacenter environment, and we use a

static hierarchy branching with RAS servers on the upper level consolidating results from all datacenters.

Our load balance algorithm is also made with regards to the managed context and fully benefits from the specific circumstances. We throw away the complexity of monitoring the load of cluster hosts. Our argument for this decision is that equal subsets of resources handled by every host, naturally leads to even work load of those hosts. To prove this, we should mention the **small granularity** of the distributed objects. Requests for a single resource go to one and same cluster host, but requests upon one resource can not load noticeably its responsible host. The probability for a set of requested resources to be handled by the same responsible host is practically zero, since every resource is mapped to a chaotic integer that has equal chances to fall upon every live host in the cluster. With this small granularity of distributed objects (resources), we can claim that we have a near-to-the-zero probability for some cluster host to become more overloaded than its neighbors.

6. Conclusions and feature work

The first achievement of our work is the definition of a hybrid distributed topology – a result of chasing after the best balance between scalability and performance efficiency for the Cloud. Next, we define a load balance algorithm that is fully calculation-based and appears to be efficient for our requirements. It does not impose overhead for data synchronization messages or shared-memory access, which extends the limits of scalability of the cluster. After that, we ensure a single handler for every possible resource, which prepares the ground for additional efficiency of using non-replica caches by resource handler modules. And finally, we achieve minimized rebalance redirections enabling an efficient failover and reliability.

A matter of future work would be to provide mathematical proof for the efficiency of the proposed scheme. Parameters should be applied for a set of characteristics - the possible host states including lightly-loaded and heavily-loaded hosts, the probable load upon the system from administration side and from resource side, etc. Calculations of probabilities to have even load and comparison of the cluster utilization in each case would mathematically prove the properness of our decisions.

Another direction of research would be to modify the algorithm for heterogeneous clusters, i.e. composed of machines with different hardware capabilities that may handle different volumes of work. This is not a practical requirement for the Cloud, since Clouds are intentionally built and hence - dedicating a homogeneous cluster is not an issue. But our algorithm could still be modified to work for a heterogeneous cluster providing good utilization without monitoring the load of every machine.

References

1. K. Krauter, R. Buyya, M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing", *Software: Practice and Experience*. vol. 32, 2, 2002, pp. 135-164
2. H. G. Rotithor, Taxonomy of dynamic task scheduling schemes in distributed computing systems, *IEE Proceedings on Computer and Digital Techniques*, 141, No. 1 (Jan. 1994), 1–10.
3. J. Yu and R. Buyya, A Taxonomy of Workflow Management Systems for Grid Computing, *Journal of Grid Computing*, Volume 3, Numbers 3-4, Pages: 171-200, Springer Science+Business Media B.V., New York, USA, Sept. 2005.
4. Wikipedia: SPOF, http://en.wikipedia.org/wiki/Single_point_of_failure
5. S. Malik, Dynamic Load Balancing in a Network of Workstations. 95.515F Research Report, 2000.
6. Zaki, M., Li, W., Parthasarathy, S. "Customized Dynamic Load Balancing for a Network of Workstations". *Proceedings of HPDC '96*, 1996..
7. Zaki, M., Li, W., Parthasarathy, S. "Customized Dynamic Load Balancing for a Network of Workstations". *Proceedings of HPDC '96*, 1996.
8. Dandamudi, S. Sensitivity Evaluation of Dynamic Load Sharing in Distributed Systems, Technical Report TR 97-12, Carleton University, Ottawa, Canada.
9. Dandamudi, S., Piotrowski, A. "A Comparative Study of Load Sharing on Networks of Workstations", *Proceedings of the International Conference on Parallel and Distributed Computing Systems*, New Orleans, October 1997.
10. Wikipedia: Caching Algorithms, http://en.wikipedia.org/wiki/Cache_algorithms
11. Mossige, Svein, Generation of permutations in lexicographical order, in: *BIT* , ISSN 1572-9125, Vol. 10 (1. 1970), p. 74-75.
12. Knuth, Donald. *The Art of Computer Programming*, volume 3 (1973), Sorting and Searching. pp. 506–542.

Sophie 2.0 — A Platform for Reading, Writing and Publishing of Electronic Books in a Network Environment

Miloslav Sredkov, Trifon Trifonov, and Kalin Georgiev¹²

¹ Faculty of Mathematics and Informatics, Sofia University,

² Astea Solutions AD

milo@asteasolutions.com, {triffon,kalin}@fmi.uni-sofia.bg

Abstract. In this paper we present Sophie 2.0, whose aim is to act as a platform supporting the life-cycle of an electronic book. We discuss the design decisions made to aid the different actors surrounding the electronic book to create it, publish it, read it, discuss it, modify it and extend it. We explain how the architecture of Sophie 2.0 is intended to enable complex, rich and dynamic content, accessed collaboratively over the network while maintaining a simple user experience. An overview of the existing open-source libraries and technologies that have been incorporated to Sophie is presented. Some novel approaches and algorithms, which were invented specifically for the project are also noted.

1 Introduction

The Sophie project was conceived by Robert Stein with the Institute for the Future of the Book [if] as an authoring tool, which allows creation of electronic books, which goes beyond the core functionality for laying out text and images by adding rich media, interactivity, time-based events and network sharing. At the same time the project had to meet a number of non-functional requirements, which guarantee a smooth user experience, intuitive interface, availability, openness and extensibility. Aided by University of Southern California, the Mellon foundation and the MacArthur foundation, the development of Sophie 1.0 started. While the product evolved, more advanced functionalities were found desirable, such as real-time collaborative authoring and reading, versioning, embedded books, template support and basic scripting capabilities. The need for a complete redesign and rewrite of the product was realised and by the time of the first release, planning of Sophie 2.0 started [s2]. The major development was executed by Astea Solutions AD, a young Bulgarian company specialising in software for electronic publishing and working closely with Sofia University “St. Kliment Ohridski”. All authors are affiliated both with Astea Solutions and with the Faculty of Mathematics and Informatics at Sofia University.

The second release extends the vision of Sophie to an integrated and fully modular platform rather than a standalone monolithic application. A special book content format was developed to enable collaboration and version support.



Supporting a wide variety of content types was a priority, hence a number of open-source libraries were employed via a unified interface to allow access to different multimedia. Sophie 2.0 itself is distributed with the Education Community License 2.0 [ecl2] — a permissive open-source license.

The purpose of this paper is to discuss the architectural and technical solutions, which were adopted so that Sophie 2.0 could meet the requirements and compete with the most professional in the field of electronic publishing. The text is structured as follows. Section 2 presents the coarse level architecture of the Sophie platform. Section 3 discusses how different types of rich content are managed and stored in a unified format. Section 4 explains some engineering aspects of the collaborative features and Section 5 summarises the most important technical features of Sophie 2.0 as well as possibilities for further improvement.

2 Architecture of Sophie 2.0

In order to achieve modularity, the platform is developed as a set of modules. In contrast to other applications, there is no main application but instead there is a tiny launcher, which starts an OSGi [osgi] framework and loads all modules. Apache Felix [felix] is used as an OSGi implementation because of its openness and active development. Each Sophie 2.0 module is a JAR file containing an OSGi bundle. The whole set of modules is called “The Sophie 2.0 Platform”, while subsets like Sophie 2.0 Server, Sophie 2.0 Author, Sophie 2.0 Reader are referred to as editions.



Fig. 1. User interface of Sophie 2.0 Author

To facilitate development of the modules and to standardize the way modules interact with the environment and each other, a small library named `org.sophie2.core.modularity` was developed. It also serves to hide the low level details of the OSGi framework. It defines interfaces for *extensions* and *extension points* allowing modules to define them. In addition, it can be queried

about the status of the modules and provides access to the static resources which are used between modules. The library uses `org.sophie2.core.prolib` in order to automatically bind extensions to the appropriate extension points.

Although the Java platform provides rich set of functionalities in a portable manner, some of the requirements of Sophie would not be easily implemented without the use of native libraries. Audio and video decoding, embedded web browser in a page are examples of Sophie features that are implemented using native code. The two most popular technologies for using native libraries in Java are JNI [Lia02] and JNA [jna]. Both of them load the libraries in the process of the Java Virtual Machine (JVM). In order to achieve better security and better independence between the Java and the native parts, another approach was taken. The native code was compiled into standalone applications for all supported platforms (32-bit Windows, 32-bit Intel Mac OSX, 32- and 64-bit GNU/Linux). The applications would be launched by Sophie 2.0 in separate processes, and the Java part would communicate with the native part using a specified protocol with the help of `org.sophie2.base.natlib`. If an error occurs in the native part, the Java part may handle it by restarting the native process if necessary without affecting the rest of the application.

In large software applications, synchronizing state between different parts of the application and keeping invariants satisfied can become a serious burden. In order to improve the situation, recent UI-related technologies employ reactive programming [Bou06]. Examples of such technologies include Adobe Flex [flex], WPF [Nat06], JavaFX [javafx]. Since standard Java lacks such features, a custom reactive programming library `org.sophie2.core.prolib` was designed and implemented and is used extensively in the application.

Sophie 2.0 implements the classic Model View Controller design pattern [Ree78]. The model consists mainly of Sophie resources such as books, pages, frames, etc. The view uses `org.sophie2.core.prolib` to keep its state synchronized with the model. The controller consists of a set of stateless enumeration classes, containing event handling logic. A small library `org.sophie2.core.mvc` is provides infrastructure to allow controllers to handle messages fired from the views.

In order for versioning, persistence and collaboration to be implemented, all Sophie resources (such as books, pages, frames, etc.) are implemented by the `org.sophie2.base.resources` library. Each resource is presented as a hierarchy of properties. The only way to change them is to apply an undo-able change using `org.sophie2.base.resources`, thus adding an entry to the undo history, notifying other collaborators, updating the views, etc.

3 Sophie 2.0 content

Text is one of the most important types of Sophie content. Sophie can lay out styled text in various shapes. Also, the auto-chaining feature automatically replicates pages and interconnects frames to form a continuous text flow. At the time of implementation, there wasn't any suitable open source library in Java capable

of solving these tasks with acceptable quality and performance. For that reason, a custom library `org.sophie2.base.model.text` was implemented. The text was implemented as immutable structures using balanced binary trees and recursive hashing. This way various optimizations such as memoization [Mic68] could be employed easily by the text layout implementation.

The pages of Sophie books may be very demanding on graphics. All elements including video should allow manipulation in a consistent way. This includes positioning, resizing and rotating. The transparency can also be adjusted, and some basic graphic effects such as shadows should be supported. This makes it necessary for the graphics to be abstracted, and a library `org.sophie2.base.scene` was implemented for that purpose. It provides a set of classes to define a scene graph [Str93] consisting of primitive graphic elements such as images, shapes, texts in a reactive way using `org.sophie2.core.prolib`. The scene interface was implemented in several different ways: `org.sophie2.main.scene.jogl` — using OpenGL bindings for Java, `org.sophie2.main.scene.simple` — using plain Java2D, `org.sophie2.main.scene.sprites` — using various raster graphics optimizations. After a series of performance experiments, the implementation `org.sophie2.main.scene.simple` gained the highest score, so it is the one that is enabled by default and supported most actively.



Fig. 2. Timelines in Sophie 2.0

Next to text, sound and video are the most important types of Sophie content. The popular Java Media Framework [jmf] and several alternatives such as FMJ [fmj] and Jffmpeg [jffmpeg] were researched. It was decided that none of them could provide the needed functionality, including supporting precise control over the timing and appearance of frames, and supporting all most popular media formats and encodings. For that reason, a custom C++ application using the FFmpeg [ffmpeg] library and working according to the `org.sophie2.base.natlib`

protocol was implemented and compiled for all supported platforms. Due to some limitations of JavaSound API such as its unsatisfactory performance on GNU/Linux, alternative native implementations using PortAudio [Ben03] are available for playback and recording. They can be enabled and disabled with a command-line parameter.



Fig. 3. Actions in Sophie 2.0

Timelines are one of the most important ways of defining interaction in Sophie 2.0. In Sophie 1.0, timelines were imperative, meaning that at some points in time, some actions that modify the state of the model are performed. Although very powerful, this approach has drawback of making some operations such as fast-forward or rewind complex to define and implement. The authors also have difficulties predicting certain behaviours, increasing the likelihood of making mistakes. In Sophie 2.0, a declarative approach was followed. That is, timelines model the state of the properties of the objects in the scene. Internally, element properties are implemented by modelling the value as a function over time rather than storing a single value.

Links allow authors to define interactivity based on the actions of the user. Their implementation is facilitated by the `org.sophie2.core.mvc` library such that additional event handlers named link processors are implemented. Each link consists of a list of link rules. Each link rule contains a trigger (such as “element shown”, “mouse clicked”, “mouse over”, etc.) and an action (such as “go to specific page”, “hide element”, “play movie”, etc.). One special kind of action is the “invoke script” action. This allows users to invoke a defined script,

that provides custom manipulation of the state of the elements. The scripts are written in JavaScript and the Rhino Engine [rhino], along with a custom implemented DOM, is used for execution. In addition to providing interaction, the scripting can also be used for automating the work of the author.

4 Collaboration

Enhancing Sophie with collaboration capabilities was a major motivation for developing version 2.0. These features can be structured as follows:

1. sharing resources (author to author)
2. publishing books (author to reader)
3. providing live feedback (reader to author & reader to reader)
4. real-time concurrent authoring (author to author)

All collaborative actions are supported by a platform element called the Sophie 2.0 Server, which together with Sophie 2.0 Reader and Sophie 2.0 Author follows a traditional client/server model. In order to adhere to the philosophy of a simple user interface, the deployment of Sophie 2.0 Server needed to be automatic and as easy as possible, so that it is feasible to use by an inexperienced user. This was achieved by developing Sophie 2.0 Server in Java SE rather than Java EE. The server was designed as a self-contained component, which is distributed with embedded third-party libraries, rather than allow a more flexible complex configuration and dynamic linking.

The server requires two major independent modules: a database and a web server. The H2 database engine [h2] was chosen, because of its main advantages: speed and small footprint. In addition, H2 was created with the possibility of embedding in mind, which made it suitable for the needs of the project. Sophie 2.0 Server also utilises the Jetty web and application server, which is a standard choice for an embedded application server, which does not require Java EE.

Jetty is used to expose two different interfaces through a single HTTP destination. On one hand, a web-based interface was developed, which allows interacting with the server via an ordinary web browser. This enables authors to share multimedia assets between each other and to upload their Sophie books for others to download. One purpose of the web-based access is to allow users that do not have Sophie installed to browse available books and view them through an applet version of the Sophie 2.0 Reader.

Support of more advanced collaboration is facilitated by the second interface, which is an API exposed to the client components of Sophie 2.0. Thus the network-enabled functions extend beyond browsing, downloading and uploading books to real-time collaboration. One of these features are comment frames, which are effectively mini-forum components that can be freely inserted anywhere inside any book. Once the author publishes a book on the server, all readers who open the book from the server establish a channel through which the comment frames are synchronised across all currently opened instances of the book. Posting of a comment through any one of those instances immediately

propagates to other readers of the book, thus allowing a live real-time discussion. Authors can actively participate in such forums by answering to questions and remarks, or even changing the book according to requests from the audience. The social effect of such a feature is that “books become places where people meet” (Robert Stein, [if]). This uncovers an interesting alternative to traditional book publishing, where usually people pay for the right to access fixed (even though possibly interactive) content, and DRM schemes need to be in place in order to protect the authors’ copyright. However, when a community is built around the book, then the content itself can be available free of charge, and readers would pay for the possibility for social interaction, which help disseminating their own views and ideas.

Finally, Sophie 2.0 supports a real-time collaborative editing mode for books. After a book is published for the first time on a server, it can be opened simultaneously by several authors and their editing changes are immediately committed on the server and propagated to all instances of the book. This is achieved via an asynchronous communication channel and a set of versioning algorithms for Sophie content, which allow a transactional representation of atomic editing commands. Thus, independent editing operations on the same version can be executed in an arbitrary order on the server by preserving consistency of the book. Operation conflicts are automatically resolved, possibly ignoring one of the editing operations. The whole live editing process can be observed and even recorded by potential readers of the book. This feature can also be combined with comment frames, now serving as discussions regarding the creation of the book.

The nature of collaborative features requires adequate control of access. This is achieved in Sophie 2.0 by supporting fine- and coarse-level access control by being able to assign specific permissions of a user regarding a book, as well as referring to groups of users or books. Security options can be managed through the web-based interface of Sophie 2.0 Server.

5 Conclusion and future work

Sophie 2.0 is an open-source project aiming to provide users with a powerful yet easy to use instrument to author, view and share rich, interactive and dynamic content. The extreme modularity achieved via the OSGi framework allows for easy extension and modification of the core functionalities of the platform. The intuitive user interface is supported by a custom reactive programming library, which allows immediate propagation of user actions and provides a smooth authoring experience. A rich set of multimedia formats is supported by the use of native libraries, while stability is guaranteed by executing native code in separate processes outside the JVM. Sophie books are stored in a custom flexible content format supported by a set of versioning algorithms, allowing merging of concurrent changes. Collaboration is achieved via a ready-to-run standalone server component, which synchronises editing actions and user comments across all connected open instances of a Sophie book.

Sophie 2.0 was released on 15 December 2009 and is considered feature complete. It is still under active development with the main goal to improve overall performance and user perception of the software. Being a relatively recent project, Sophie 2.0 is currently in the process of gathering a user community, whose feedback will be valuable for the further development of the platform.

Acknowledgements. The authors gratefully acknowledge financial support within project No.168/19.04.2010 with the Scientific Research Fund of Sofia University “St. Kliment Ohridski”.

References

- [Ben03] Bencina, Ross. PortAudio and Media Synchronisation — It’s All in the Timing, Proceedings of the Australasian Computer Music Conference, pages 13–20, 2003, http://www.portaudio.com/docs/portaudio_sync_acmc2003.pdf
- [Bou06] Boussinot, Frederic et al. Reactive Programming Description, 2006, <http://www-sop.inria.fr/mimosa/rp/generalPresentation/>
- [Lia02] Liang, Sheng. The Java™ Native Interface Programmer’s Guide and Specification, Addison-Wesley, 2002, <http://java.sun.com/docs/books/jni/download/jni.pdf>
- [Mic68] Michie, Donald. Memo Functions and Machine Learning, Nature, No. 218, pp. 19–22, 1968.
- [Nat06] Nathan, Adam. “Windows Presentation Foundation Unleashed (WPF)”, Sams Publishing, 2006.
- [Ree78] Reenskaug, Trygve, MVC XEROX PARC 1978–79, <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>.
- [Str93] Strauss, Paul, “IRIS Inventor, a 3D Graphics Toolkit”, ACM SIGPLAN Notices, Vol. 28, No. 10, pp. 192–200, 1993.
- [ec12] Open Source Initiative, Educational Community License, Version 2.0, April 2007, <http://www.opensource.org/licenses/ec12.php>
- [felix] Apache Software Foundation, Apache Felix OSGi implementation, <http://felix.apache.org/>
- [ffmpeg] FFmpeg audio/video converter, <http://ffmpeg.org/>
- [flex] Adobe Systems Incorporated, Adobe Flex SDK, <http://www.adobe.com/products/flex/>
- [fmj] Freedom for Media in Java (FMJ), <http://fmj-sf.net/>
- [h2] H2 Database Engine, <http://www.h2database.com/>
- [if] Institute for the Future of the Book, <http://www.futureofthebook.org/>
- [jmf] Oracle, Java Media Framework, <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>
- [javafx] Oracle, JavaFX, <http://javafx.com/>
- [jffmpeg] Jffmpeg, a codec pack for the Java Media Framework, <http://jffmpeg.sourceforge.net/>
- [jna] Java.net, Java Native Access (JNA), <https://jna.dev.java.net/>
- [osgi] OSGi Alliance, OSGi Service Platform Release 4, Version 4.2, <http://www.osgi.org/Download/Release4V42>
- [rhino] Mozilla Foundation, Rhino: JavaScript in Java, <http://www.mozilla.org/rhino/>
- [s2] Sophie 2.0 Project Page, <http://sophie2.org/>

Grid framework for e-learning services

Magdalena Todorova

“St. Kliment Ohridski” University of Sofia, Faculty of Mathematics and Informatics
Sofia 1164, Bulgaria
magda@fmi.uni-sofia.bg

Abstract. E-learning, grid computing and their integration are issues which attract a growing interest during the last decades. This paper presents a grid framework for e-learning services called GridEdu. The framework is based on several new technologies such as: grid computing, learning management systems, Web services, ontology. The paper briefly describes the technologies used, the architecture and the components of the GridEdu, the integration of the components.

Keywords: Grid Computing, E-learning, Learning Management System, Web Service, Ontology

1 Introduction

E-learning can be defined as a “wide set of applications and processes, which use available electronic media (and tools) to deliver vocational education and training” [1]. The wide use of the Internet led to the growing interest in e-learning. With the convenience of wireless and mobile technologies, it is easily available to be applied anytime, anywhere, any way. As a result, many e-learning platforms and systems have been developed. These are based on client-server [2], [3], on peer-to-peer [4], [5], on Web service architectures [6], [7], and on hybrid architectures [8]. According to Screen Digest, it is estimated that the U.S. Corporate e-Learning market will reach 50 billion USD by 2010 [9]. This market was about 3,5 billion USD in 2007/2008. A Grid-gateway „e-L&e-S” for e-learning and e-science is planned to be built as an experiment in Russia. Its goal is “to integrate the e-learning technologies with Grid computing so to support the Russian national system, IT-education and vocational training of elite science and technology professionals in the field of science and high technologies” [10].

There exists a certified Grid farm at Sofia University (SU) since 2006, which is included in the European Grid. The main task in front of the SU Grid farm is to develop a software and hardware demonstrative infrastructure which to support the demonstration of the advantages of the meta computing in solving tasks related to:



processing large amounts of data located in different parts of the world; collaborative and effective use of computing resources, organized in different heterogeneous and widely distributed structures; real-time remote access to unique tools; modeling a virtual reality via the interface. So far, the Grid farm at SU is used for educational purposes.

The current article presents a brief description of a hybrid architecture of an e-learning platform. The architecture has been developed by Grid computing, e-learning systems and technologies and Web services. It is about to be implemented for educational purposes at the Faculty of Mathematics and Informatics (FMI) at SU.

2 Fundamentals

This part contains a brief description of the used in GridEdu systems and technologies.

2.1 Grid Computing

The fundamental work [11], 1998, by the inventors of the Grid technologies Ian Foster (Argonne National Laboratory) and Carl Kesselman (Institute of Information Technologies, University of South California) gives the following definition of Grid: “A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities.” This definition was further narrowed and clarified during the years to follow. For example, in order to overcome social and political problems, the authors specify [12] that Grid computing is concerned with “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations”.

More detailed explanation of the given definitions can be found in [12] and with a short introduction to the concepts in [13]. Further discussion on Grid standardization efforts is given in [14] and in [16] are discussed the standardization efforts for implanting of Grid computing in the enterprises. Very important aspects for the success of Grid computing is Grid governance discussed in [17] and Service-Oriented Architecture discussed in [15] as software implementation of the Grid.

Grids are typically implemented as a form of middleware which provides all grid-related services and can also use the Internet as a communication infrastructure. The main components of a Grid are: grid fabric, core grid middleware, and grid application (fig. 1) [12].

- *Grid fabric*

This component consists of all globally distributed resources that are accessible anywhere via the Internet. These resources could be: computational resources, storage resources, network resources, code repositories, catalogues, sensors.

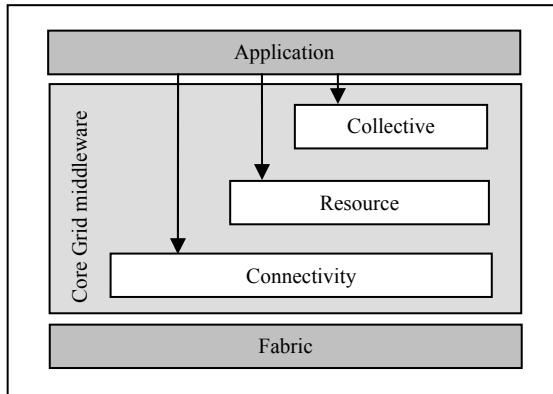


Fig. 1. The layered Grid architecture

- *Core Grid middleware*

This component offers core services such as remote process management, co-allocation of resources, storage access, information registration and finding, security, and aspects of Quality of Service. It is organized in three layers: connectivity, resource and collective ones.

The *connectivity* layer defines main communication and security protocols required for Grid-specific network transactions. The communication protocols allow for data transfer between the fabric layer resources. The security protocols offer cryptographic security mechanisms, through which the identification of the resources and users is checked.

The *resource* layer implements access to single resources and monitors their status. It defines protocols for security negotiation, monitoring, control, accounting, and payment of sharing operations on individual resources. Two resource layer protocols exist: *information* protocols (which give information on the status and the structure of single resources) and *management* protocols (negotiate the access to the distributed resources, specify the resource requirements, and point to the operations which have to be performed).

The *collective* layer coordinates multiple resources. This layer contains protocols and services that are not associated with any specific resource. Its protocols and services are global in nature and coordinate interactions among collections of resources.

- *Grid application*

This layer includes the user applications that operate within a virtual organization environment. These applications are typically developed using Grid-enabled languages and utilities.

The contemporary developments in distributed virtualization frameworks lead to a noticeable transfer of grid technologies to the cloud frameworks. Although its technological similarity these two paradigms reveal some important differences. Whereas grids serve as a mean for unifying resources of various owners/administrators in a virtual organization (VO) the cloud is viewed as a virtual distributed server with a unified ownership and administration (VS) [18]. Our research is focused on the grid frameworks as a practical way of sharing resources data and services in an integrated e-learning framework.

2.2 E-learning

“The e-learning is an attribute of the global information society and has developed in response to the needs of the contemporary student of flexible education available anytime anywhere. It is an education which requires information and communication technologies and Internet for its design, implementation and management” [19]. The e-learning systems are divided into: Learning Management System (LMS), Learning Content Management System (LCMS), Virtual Learning Environment (VLE), Content Management System (CMS), etc. The LMS are software packages which support the delivery and management of learning materials to large groups of learners. LCMS are at a higher level than LMS and combine authoring tools and LMS through which these resources reach the end user. The VLEs are software systems developed to facilitate the trainers to manage the disciplines. The CMSs are developed to facilitate the publishing and management of a large volume of learning resources.

The GridEdu architecture presented in this paper uses an LMS. Most models for e-learning architecture consist of 4 parts (Fig. 2).

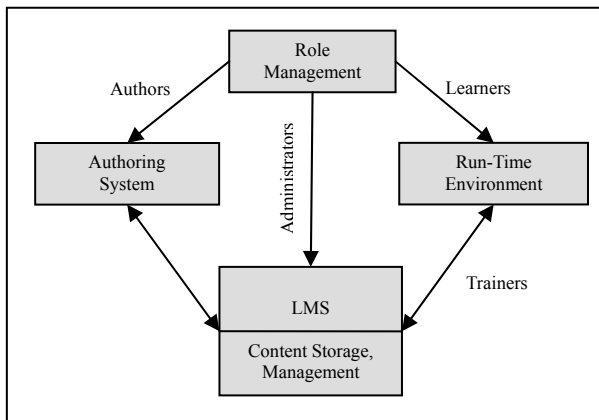


Fig. 2. Convention model for e-learning architecture

The components: Authoring system, Run-time Environment and Learning Management System can be logically and physically distributed. An LMS offers: management of the educational process; material delivery; course administration; course delivery interface; databases (DB) for learners' work storage; a system for design, storage, selection and delivery of personalized learning contents in the form of standardized Learning Objects (LO) [20]. The main roles in an LMS are: authors (create contents stored under the control of the LMS in a DB), learners (consume content), trainers and administrators. The learning objects offer content units, exercises, and tasks to be performed and used in a single session.

2.3 Web Services

Web services are application programming interfaces that are accessed via Hypertext Transfer Protocol (HTTP). The Internet based protocols for web-services include Web Service Description Language (WSDL), Simple Object Access Protocol (SOAP), and Extensible Markup Language (XML).

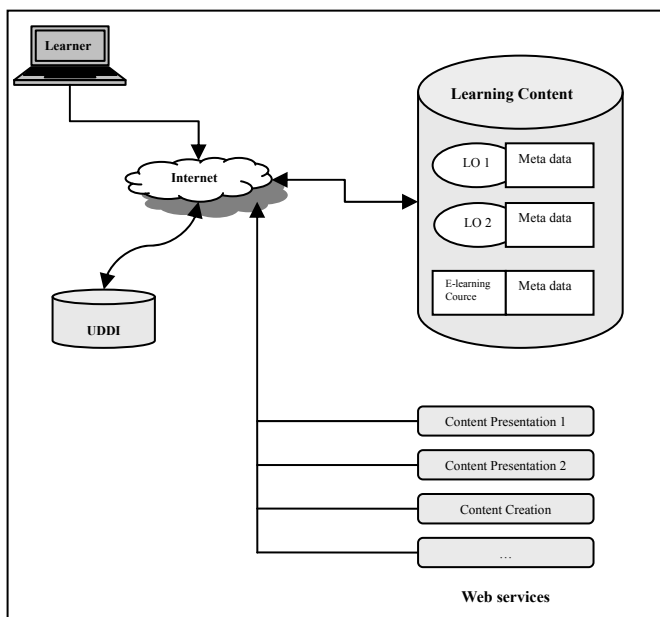


Fig. 3. LMS as a Web Service

The core of the Web service technology is the WSDL. It is an XML-based language, which contains elements to complete definition of Web services. WSDL creates a

framework for describing web services based on service protocols. SOAP is responsible for the relation between the consumer and the provider. It has a mechanism for sending and receiving messages and is compatible with SMTP, FTP and HTTP protocols. The Universal Description, Discovery and Integration (UDDI) protocol is used for finding new services and publishing existing ones. UDDI contains a standard description of services.

E-learning can be presented as a collection of Web services which process learning contents and disciplines in the same way as an LMS (fig. 3).

2.4 Ontology

„In computer science, an ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain, and may be used to describe the domain. In theory, an ontology is a formal, explicit specification of a shared conceptualization” [21]. Education assigns an important role to the ontologies as they allow for specifying knowledge such as concepts, relations, classification and reasoning.

Applying ontology to education brings the following main advantages [22]:

- It makes systems smarter and more reflexive.
- It explicates the conceptualization on which a system is based.
- It contributes to standardization of vocabulary.
- It enables them to be literate and hence to communicate with humans.
- It makes knowledge reusable, and so forth.

An educational ontology describes:

- *all activities and operations of education and the relations between them*

The language OWL for Services (OWL-S) offers an adequate environment for the realization of this part of the ontology. It supplies a core set of ontology concepts for describing the properties and capabilities of Web services in unambiguous, computer-interpretable form.

- *the educational content organization*

This part of the ontology can be realized via the Web Service resource model.

3 A Grid platform for e-learning

In this section the GridEdu architecture is outlined. It contains: Education Service Grid, Education Service LMS and e-Learning Grid Portal (fig. 4). The Education Service Grid is based on grid services. The Education Service LMS is based on Web services and Education Ontologies. The GridEdu portal maintains the connection between the learners and the GridEdu modules.

3.1 Education Service Grid

This component (fig. 5) represents the layers of the Grid architecture (see fig.1), adapted for e-learning. The fabric layer is a Java applet which supplies a unified interface for all resources in the Education Service Grid.

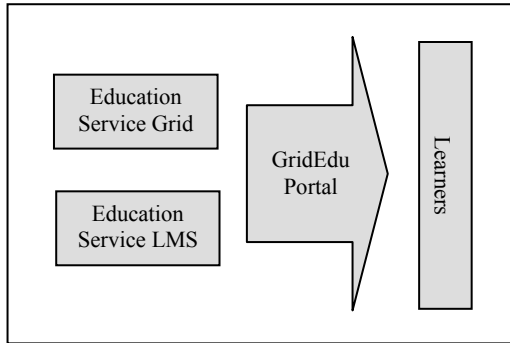


Fig. 4. GridEdu architecture

The connectivity layer specifies a grid login service. The service has the following main functions: to create new users; to delete users; to register computers; to de-register computers, to login Grid, to logout Grid, etc.

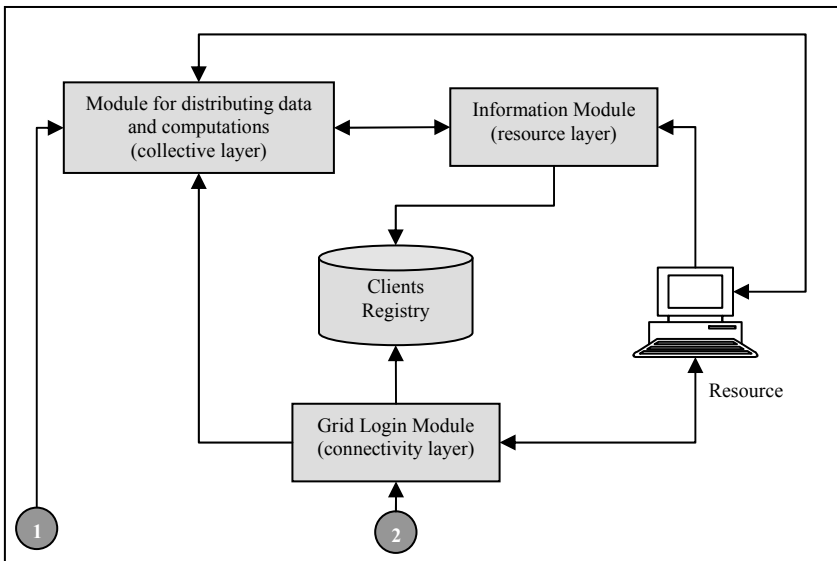


Fig. 5. Education Service Grid

A symmetric cryptographic algorithm is used for implementation of the login service. The resource layer specifies an information service which determines the status and type of all resources in the grid. Firstly, it determines which computers are available. The information service can do the following major operations:

- find information for the status of each computer;
 - find information for the resource type of each computer;
 - find the restrictions used for each computer;
 - generate a list of information on the status of all computers;
- etc.

The collective layer is responsible for distributing data and computations in the Education Service Grid. For the distribution of data it implements a functionality that is typical of data grids. The service can do the following major operations: registration; copying; deletion; data searching; etc. It could be realized through a grid scheduling algorithm [23].

3.2 Education Service LMS

Education Service LMS functions are to coordinate all education related activities. Its architecture (fig. 6) is similar to the one of a traditional learning management system organized as a Web service (fig. 3). Its functionality is realized as Web services. It offers both content, which uses Education Service Grid as well as content, which does not use Education Service Grid. It consists of the following elements: LMS Login Module, LMS Course Manager Module, Education Ontology Module, Content Module, Authoring Module, and Additional Module.

- *LMS Login Module*

This component performs the login service. It is a functionality which allows for user registration and performance of some adaptation transformations.

- *LMS Course Manager Module*

This component performs the course management services. They support the course management. The learners can search for courses, identified as learning objects. They can enroll for courses, attend lessons, etc.

- *Education Ontology Module*

This component performs ontology services. These services support a semantic search for courses and provide operations such as insertion, deletion, edition and searching for elements in the ontology. New content units can be added in order to improve the contents.

- *Content Module*

This component performs services which provide e-learning content. The learning content consists of the following parts: a learning object, an assessment part and metadata. A learning object is a lesson or part of a lesson in HML or HTML format. The assessment part consists of online tests through which the learners or their tutors

can check to what extent the contents have been learnt. The metadata part consists of information for searching the computers containing e-learning contents. The content module also contains Grid Learning Objects (GLOBs). This is the way the Education Service LMS uses the Education Service Grid functionality.

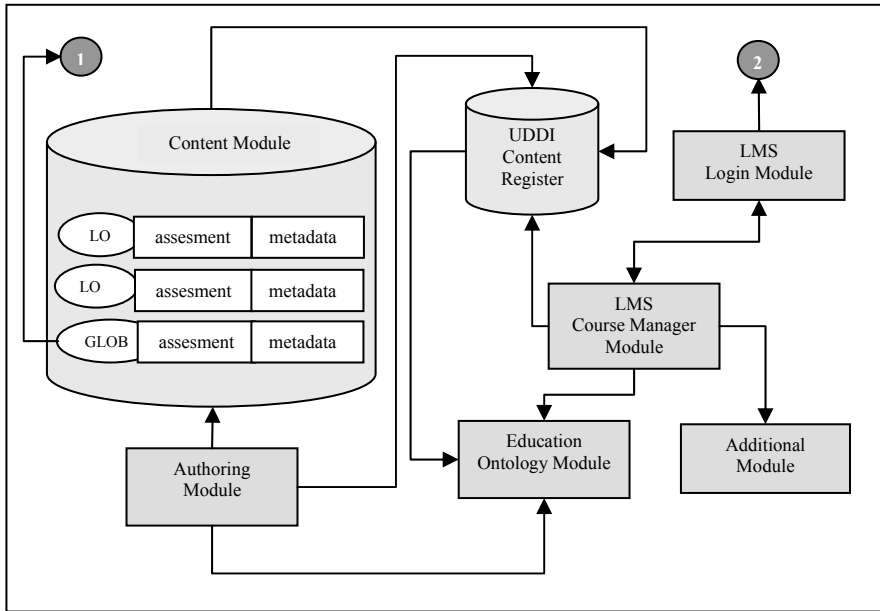


Fig. 6. Education Service LMS

- *Authoring Module*

This component offers authoring services. These services allow for creating, editing and publishing e-learning contents so it to be found by the Education Service LMS.

- *Additional Module*

This component can be used for other services as chat, discussion, assessment results, accounting, etc.

3.3 Integration of Education Service Grid and Education Service LMS

The learner connects to the GridEdu via an e-learning PC. Through the login service of the Education Service LMS, the e-learning PC becomes a resource of the Education Service Grid. After the learner has registered, the fabric layer applet of the Education Service Grid can be transferred as code and to be executed by the e-learning PC. This supports the communication with the Education Service Grid. The login service of the Education Service LMS refers to the grid login service and executes the operation of

loading parameters in the Grid: the learner's data and their PC data. Thus the e-learning PC authentication in the Education Service Grid is performed. If the user has not registered in the grid login service, the Education Service LMS login service can initiate the creation of a new Grid user.

GridEdu component integration is performed also via the use of GLOBs.

A GLOB enhances the traditional learning object functionality by adding a grid functionality, which consists of specific grid application layer and user interface (fig. 1). The design of a GLOB has been borrowed from [24] and contains the following components: an overview of the lesson, metadata, re-usable information objects (RIOs), and a summary. The metadata is used for finding GLOBs. The RIO consists of: contents, practice items and assessment items. The content part contains learning material for the educational part. The contents can either be saved in a database or in XML or HTML format. The practice items are used to generate online exercises for the learners, and the assessment items are used to generate online tests for the final exam. An RIO could contain additional grid functionality which is realized as a grid application layer functionality and is accessed via the user interface.

4 Conclusion

The e-learning services framework presented here is a complicated engineering work and is about to be implemented and experimented on. Through using this Grid learning environment we expect results which will lead to improved IT education and will support the training of elite professionals in the field of sciences and high technologies.

Acknowledgements. The paper is supported by Grant 162/2010 of Sofia University Research Fund.

References

1. Eklund, J., Kay, M., Lynch, H. M.: "E-Learning: Emerging Issues and Key Trends: A Discussion Paper", Australian National Training Authority (ANTA), 2003, available at http://flexiblelearning.net.au/research/2003/elearning2509_03final.pdf
2. Brzoza, P.: E-learning platform for interactive access to multimedia materials in daisy format, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-415/paper_16.pdf (last visited 30.06.2010)
3. Shoikova, E., Ivanova, M.: Learning Design Implementation in SCORM E-Learning Environment, <http://dspace.ou.nl/bitstream/1820/849/1/Paper32.pdf> (last visited 30.06.2010)
4. Jin H., Yin Z.: APPLE: A Novel P2P Based e-Learning Environment, LNCS, 3326, pp 52-62 (2004)

5. Kawamura, T., Kinoshita, Sh., Sugahara, K.: A MOBILE AGENT-BASED P2P E-LEARNING SYSTEM, Proceedings of the 16th Lusted International Conference Parallel and Distributed Computing and Systems, Nov. 9-11 (2004)
6. Qiu, X., Jooloor, A.: Web Service Architecture for e-Learning, <http://grids.ucs.indiana.edu/ptliupages/publications/E388NH.pdf> (last visited 30.06.2010)
7. Jabr, M., Al-Omari, H., e-Learning Management System Using Service Oriented Architecture, *Journal of Computer Science* 6 (3): 285-295 (2010)
8. Meguro, K., Motomura, Sh., Kawamura, T., Sugahara, K.: Distributed e-Learning System with Client-Server and P2P Hybrid Architecture, *World Academy of Science, Engineering and Technology* 55 2009, <http://www.waset.org/journals/waset/v55/v55-47.pdf> (last visited 30.06.2010)
9. Kaliski, J., Kalinowski, J., Schumann, P., Scott, T., Shin, D.: Competition in the e-Learning Industry: A Case Study, *Journal of Business Case Studies*, Volume 4, Number 2 105 (2008)
10. A Grid-gateway „e-L&e-S” for e-learning and e-science, <http://www.dubna.ru/34/2882.html> (last visited 30.06.2010)
11. Foster, I., Kesselman, C.: *The Grid, Blueprint for a New computing Infrastructure*, Morgan Kaufmann Publishers, Inc., 1998. I. Foster and C. Kesselman (eds), *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers (2004)
12. Foster, I., Kesselman, C., Tuecke, S.: *The Anatomy of the Grid. Enabling Scalable Virtual Organizations*, <http://www.globus.org/alliance/publications/papers/anatomy.pdf>, (2000), (last visited 30.06.2010)
13. Dimitrov, V.: Introduction of Grid, *Proc. Of Int. Scientific Conf. „Information technology in business management”*, 16-17 oct. 2009, Varna, pp 420-425 (2009)
14. Dimitrov, V.: Grid and Standardization Efforts, *Proc. of Int. Scientific Conf. Unitech’08*, 21-22 November 2008, Gabrovo, pp. I-296 – I-300 (2008)
15. Dimitrov, V.: Service-oriented architecture, *Technologica*, 2009, 208 pages
16. Dimitrov, V.: Enterprise Grid Computing, *Proc. of Fourth Int. Bulgarian-Greek Conf. Computer Science’2008*, Kavala, 18-19 September 2008, pp. 737-742 (2008)
17. Dimitrov, V.: Grid Governance, *Third International Conference on Information Systems & Grid Technologies*, 28-29 May 2009, Sofia, pp. 126-130 (2009)
18. Zhelev, R., Georgiev, V.: Service Functionality for Cloud Systems, *Third International Conference on Information Systems & Grid Technologies*, 28-29 May 2009, Sofia, pp. 168-182 (2009)
19. Kovatcheva, E.: <http://elearn.uni-sofia.bg/course/> (last visited 30.06.2010)
20. Kovatcheva E. & Roumen N.: Design of an e-Learning System for Accreditation of Non-Formal Learning, ed. Dr Dan Remenyi, *Proceeding of the 7th European Conference on e-Learning*, Agia Napa, Cyprus, 6-7 November 2008 , book2 pp. 1-7 (2008)
21. <http://en.wikipedia.org/wiki/Ontology> (last visited 30.06.2010)
22. Mizoguchi, R., Ikeda, M., Sinita. K.: Roles of Shared Ontology in AI-ED Research – Intelligence, Conceptualization, Standardization, and Reusability – <http://www.ei.sanken.osaka-u.ac.jp/pub/miz/miz-ai-ed97.pdf>, (1997)
23. Hamscher, V., Schwiigelshohn, Streit, A., Yahyapour, R.: Evaluation of Job-Scheduling Strategies for Grid Computing, *LNCS 1971*, Springer-Verlag, Berlin, pp.191-202 (2000)
24. Barrit, C.: *CISCO systems Reusable Learning Object Strategy – Designing Information and Learning Objects Through Concept, Fast Procedure, Process, and Principle Templates*, Version 4.0, White Paper (2001)

Toward Semantic Specification of WS-BPEL

Vladimir Dimitrov¹,

¹ University of Sofia, Faculty of Mathematics and Informatics, James Bourchier Blvd. 5,
1164 Sofia, Bulgaria
cht@fmi.uni-sofia.bg

Abstract. WS-BPEL is a standard for specification of business processes composed of services. This standard is not formally defined, because it has to be opened for many different implementations. It is a XML-based standard, that is syntactically specified as mark up notation, but its semantics is informally described. This paper presents an approach to formal semantic specification of WS-BPEL. The approach is based on language constructs that are important for the essence of the standard.

Keywords: formal specification, WS-BPEL, semantic.

1 Approach Overview

WS-BPEL [1] plays a special role in WS-* specifications. Its intention is to be used for development of composed Web services. At the basis, Web services have to be implemented in some programming language (or wrap legacy software); at the recursion, Web services are composed Web services from basic Web services or other composed Web services. WS-BPEL permits multilevel hierarchy of services to be created. That is why, it is important for every Service-Oriented Architecture (SOA) [2] platform to support WS-BPEL in an orchestration service.

WS-BPEL has been originally developed by IBM and Microsoft as BPEL4WS that has been standardized as WS-BPEL by OASIS. It is a XML-based language and is open for extensions.

With WS-BPEL, as is mentioned in [3], service static composition can be specified and more precisely: service connectivity, exception handling and compensation.

There are several groups of formal methods applicable for service composition specification: automata, Petri nets and process algebras. The last two, Petri nets and process algebras, cover fully functionality given by WS-BPEL, which means that they are suitable for formal specification of WS-BPEL code. There are several papers devoted to formal semantic and analyses of control flow in WS-BPEL using Petri nets.

Our approach is based on the process algebra: Communicating Sequential Processes (CSP) [4]. It covers all needed functionality for WS-BPEL and has no sophisticated construction that makes presentation unclear.

The first step is to clarify WS-BPEL constructs in a simple programming language - BPELs. WS-BPEL is XML-based language, that is more machine readable than



human one. BPELs is useful not only for research purposes; it can be used for education too.

The next step is to present how the basic constructs of this simplified language can be converted to CSP constructs. This means that a specification in WS-BPEL (simplified) will be available for verification.

In this paper, this WS-BPEL simplified version is presented.

2 BPELs

Let's first represent how looks a simple program in WS-BPEL (this example is taken from WS-BPEL documentation):

```
<process name="purchaseOrderProcess"
  targetNamespace="http://example.com/ws-bp/purchase"
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/
    process/executable"
  xmlns:lns="http://manufacturing.org/wsd/purchase">
```

WS-BPEL specification is embraced by `<process ...>` and `</process>`. A process with name `purchaseOrderProcess` is specified. The other attributes of this tag defines `targetNamespace` that is with URI:

`http://example.com/ws-bp/purchase`

to which belong specified process. The next attribute `xmlns` tell us

`http://docs.oasis-open.org/wsbpel/2.0/ process/executable`

that Executable Process for WS-BPEL v2.0 namespace is used. The third attribute specify, that another XML name space is used with the name `lns` from URI:

`http://manufacturing.org/wsd/purchase`

This name space contains WSDL specifications of services used in the example. They are given for reference in the appendix and we will not discuss them further. Type defined in this space have to be prefixed by `lns:`.

In BPELs:

```
process purchaseOrderProcess (
  targetNamespace="http://example.com/ws-bp/purchase",
  xmlns="http://docs.oasis-open.org/
    wsbpel/2.0/process/executable",
  xmlns:lns="http://manufacturing.org/wsd/purchase") {
...
}
```

Process attributes are enclosed in `(...)`, and process content is enclosed in `{ ... }`. Attributes, here, include name spaces that are used later directly as types in the program. The WS-BPEL tags are used as keywords in BPELs.

The next piece of code is comment, which is given in English:

```
<documentation xml:lang="EN">
  A simple example of a WS-BPEL process for handling
  a purchase order.
</documentation>
```

In BPELs, the comment is:

```
// A simple example of a WS-BPEL process
// for handling a purchase order.
```

Comments are like in C++.

Then follows partner links section:

```
<partnerLinks>
  <partnerLink name="purchasing"
    partnerLinkType="lms:purchasingLT"
    myRole="purchaseService" />
  <partnerLink name="invoicing"
    partnerLinkType="lms:invoicingLT"
    myRole="invoiceRequester"
    partnerRole="invoiceService" />
  <partnerLink name="shipping"
    partnerLinkType="lms:shippingLT"
    myRole="shippingRequester"
    partnerRole="shippingService" />
  <partnerLink name="scheduling"
    partnerLinkType="lms:schedulingLT"
    partnerRole="schedulingService" />
</partnerLinks>
```

This section defines the different parties that interact with the business process in the course of processing the order. The four definitions shown here correspond to the sender of the order (customer) purchasing, as well as the providers of price (invoicing provider) invoicing, shipment (shipping provider) shipping, and manufacturing scheduling services (scheduling provider) scheduling. Each partner link is characterized by a partner link type and either one or two role names. This information identifies the functionality that must be provided by the business process and by the partner service for the relationship to succeed, that is, the port types that the purchase order process and the partner need to implement.

In BPELs, links section is:

```
partnerLinks
  lms:purchasingLT purchasing(myRole purchaseService);
  lms:invoicingLT invoicing(myRole invoiceRequester,
    partnerRole invoiceService);
  lms:shippingLT shipping(myRole shippingRequester,
    partnerRole shippingService);
  lms:schedulingLT scheduling
    (partnerRole schedulingService);
```

In link definition, the name of the link is preceded by its type, and then in parentheses the used roles are defined preceded by keyword mentioning the role type. If both roles are given, then they separated by comma. The roles order is not important. Here, the types are used directly as they are include by process attributes.

The next section defines the data variables used by the process, providing their definitions in terms of WSDL message types, XML Schema types (simple or

complex), or XML Schema elements. Variables allow processes to maintain state between message exchanges.

```
<variables>
  <variable name="PO" messageType="lns:POMessage" />
  <variable name="Invoice"
    messageType="lns:InvMessage" />
  <variable name="shippingRequest"
    messageType="lns:shippingRequestMessage" />
  <variable name="shippingInfo"
    messageType="lns:shippingInfoMessage" />
  <variable name="shippingSchedule"
    messageType="lns:scheduleMessage" />
</variables>
```

There are five variables: PO, Invoice, shippingRequest, shippingInfo, and shippingSchedule with corresponding types from lns name space: POMessage, InvMessage, shippingRequestMessage, shippingInfoMessage, and scheduleMessage. They are used to store process state.

In BPELs, variables section is:

```
variables
  lns:POMessage PO;
  lns:InvMessage Invoice;
  lns:shippingRequestMessage shippingRequest;
  lns:shippingInfoMessage shippingInfo;
  lns:scheduleMessage shippingSchedule;
```

The definition of variable is like in C++ - variable name is preceded by its type.

The next section contains fault handlers defining the activities that must be performed in response to faults resulting from the invocation of the assessment and approval services. In WS-BPEL, all faults, whether internal or resulting from a service invocation, are identified by a qualified name. In particular, each WSDL fault is identified in WS-BPEL by a qualified name formed by the target namespace of the WSDL document in which the relevant port type and fault are defined, and the NCName of the fault.

```
<faultHandlers>
  <catch faultName="lns:cannotCompleteOrder"
    faultVariable="POFault"
    faultMessageType="lns:orderFaultType">
    <reply partnerLink="purchasing"
      portType="lns:purchaseOrderPT"
      operation="sendPurchaseOrder" variable="POFault"
      faultName="cannotCompleteOrder" />
  </catch>
</faultHandlers>
```

There is one fault handler that catches cannotCompleteOrder from lns name space. This handler accepts fault data in variable POFault of type lns:orderFaultType. Its reaction is to reply to partner link purchasing - the response indicates a fault, invoking the operation sendPurchaseOrder of type lns:purchaseOrderPT and delivers variable POFault contents with fault name cannotCompleteOrder.

In BPELs, fault handlers section is:

```
faultHandlers
```

```

catch (faultName lns:cannotCompleteOrder,
      faultVariable lns:orderFaultType POFault) {
  reply (partnerLink purchasing,
        portType lns:purchaseOrderPT,
        operation sendPurchaseOrder,
        variable POFault, faultName cannotCompleteOrder);
};

```

Every catch is defined separately in this section. Catch attributes are enclosed in parentheses (...), and catch activity in curly parentheses { ... }. As activity, here, reply is introduced, port type is not obligatory, but it is put to increase readability.

Next is normal execution activity sequence, that means the activities between <sequence> and </sequence> are executed one after one:
 <sequence>

In BPELs, sequence is an operator that encloses activity, which will be executed in the given order one by one:
 sequence { ... }

The first activity in the sequence is receive:
 <receive partnerLink="purchasing"
 portType="lns:purchaseOrderPT"
 operation="sendPurchaseOrder"
 variable="PO" createInstance="yes">
 <documentation>
 Receive Purchase Order
 </documentation>
 </receive>

When from the partner link purchasing is received a message from the operation sendPurchaseOrder of type lns:purchaseOrderPT in variable PO, then a new instance of the process is created.

In BPELs, receive activity is as operator with all parameters enclosed in (...):

```

// Receive Purchase Order
receive (partnerLink purchasing,
        portType lns:purchaseOrderPT,
        operation sendPurchaseOrder,
        variable PO, createInstance yes);

```

The second activity in the sequence is flow, which means that activities embraced by <flow> and </flow> are executed in parallel:
 <flow>

The flow is commented:

```
<documentation>
  A parallel flow to handle shipping, invoicing
  and scheduling
</documentation>
```

In BPELs, the flow operator encloses activities that will be executed in parallel:
flow { ... }

Here are links used for synchronization purposes among activities in the flow:

```
<links>
  <link name="ship-to-invoice" />
  <link name="ship-to-scheduling" />
</links>
```

In BPELs, it is the first section that simply defines synchronization links, that have to be comma separated:

```
links ship-to-invoice, ship-to-scheduling;
```

This is the first parallel activity in the flow. It is a sequence of activities:

```
<sequence>
```

In BPELs, again sequence operator:

```
sequence { ... }
```

The first activity in the nested sequence is assign:

```
<assign>
  <copy>
    <from>$PO.customerInfo</from>
    <to>$shippingRequest.customerInfo</to>
  </copy>
</assign>
```

It simply copies from variable element \$PO.customerInfo to variable element \$shippingRequest.customerInfo.

In BPELs, a simplified form of assign operator is used:

```
$shippingRequest.customerInfo := $PO.customerInfo;
```

The next activity is to invoke partner link shipping on operation requestShipping of port type lns:shippingPT delivering the contents of variable shippingRequest and waiting to receive result in variable shippingInfo. This is synchronized communication:

```
<invoke partnerLink="shipping"
  portType="lns:shippingPT"
  operation="requestShipping"
  inputVariable="shippingRequest"
  outputVariable="shippingInfo">
  <documentation>Decide On Shipper</documentation>
  <sources>
    <source linkName="ship-to-invoice" />
  </sources>
</invoke>
```

This activity is commented and with source link ship-to-invoice from enclosing flow. This means that after the execution of this activity the activity staying on the target of the link could be executed in parallel. In the case, it is invoke activity below in the next sequence: second activity – search in the target container the same link.

In BPELs, it is operator invoke:

```
// Decide On Shipper
```

```

invoke (partnerLink shipping,
       portType lns:shippingPT,
       operation requestShipping,
       inputVariable shippingRequest,
       outputVariable shippingInfo,
       source ship-to-invoice);

```

The source section of WS-BPEL is here as parameter.

The last activity in this sequence is receive:

```

<receive partnerLink="shipping"
         portType="lns:shippingCallbackPT"
         operation="sendSchedule"
         variable="shippingSchedule">
  <documentation>Arrange Logistics</documentation>
  <sources>
    <source linkName="ship-to-scheduling" />
  </sources>
</receive>

```

In BPELs, it is operator receive:

```

// Arrange Logistics
receive (partnerLink shipping,
        portType lns:shippingCallbackPT,
        operation sendSchedule,
        variable shippingSchedule,
        source ship-to-scheduling);

```

In this operator source section is parameter too.

It waits invoking partner link shipping on operation sendSchedule of type lns:shippingCallbackPT and receive the result in variable shippingSchedule. This activity is documented and is a source to the synchronization link ship-to-scheduling.

Here is the end of first activity in the flow:

```

</sequence>

```

In BPELs, it is simply:

```

};

```

The second activity in the flow is again sequence:

```

<sequence>

```

In BPELs, again operator sequence:

```

sequence { ... }

```

It's first activity is invoke of partner link invoicing on operation initiatePriceCalculation of port type lns:computePricePT and receiving the result in variable PO – does not deliver data to its partner. This activity is commented:

```
<invoke partnerLink="invoicing"
  portType="lns:computePricePT"
  operation="initiatePriceCalculation"
  inputVariable="PO">
  <documentation>
    Initial Price Calculation
  </documentation>
</invoke>
```

In BPELs, it is operator invoke:

```
// Initial Price Calculation
invoke (partnerLink invoicing,
  portType lns:computePricePT,
  operation initiatePriceCalculation,
  inputVariable PO);
```

The next activity is invoke of partner link invoicing on operation sendShippingPrice of port type lns:computePricePT (it is optional and is put here to improve readability) delivering contents of variable shippingInfo. This activity is commented and is target of the link ship-to-invoice:

```
<invoke partnerLink="invoicing"
  portType="lns:computePricePT"
  operation="sendShippingPrice"
  inputVariable="shippingInfo">
  <documentation>
    Complete Price Calculation
  </documentation>
  <targets>
    <target linkName="ship-to-invoice" />
  </targets>
</invoke>
```

In BPELs, again operator invoke, but with targets as parameter:

```
invoke (partnerLink invoicing,
  portType lns:computePricePT,
  operation sendShippingPrice,
  inputVariable shippingInfo,
  target ship-to-invoice);
```

The last activity in this sequence is receive activity on the partner link invoicing with operation sendInvoice. The result is received in variable Invoice.

```
<receive partnerLink="invoicing"
  portType="lns:invoiceCallbackPT"
  operation="sendInvoice"
  variable="Invoice" />
```

```

    In BPELs, operator receive:
receive (partnerLink invoicing,
    portType lns:invoiceCallbackPT,
    operation sendInvoice, variable Invoice);
    This is the end of second sequence:
</sequence>
    In BPELs, simply:
};
    This is the third activity in the flow – it is again sequence:
<sequence>
    In BPELs, operator sequence:
sequence { ... }
    The next activity is invoke of partner link scheduling of operation
requestProductionScheduling of port type lns:schedulingPT and
with delivered contents of variable PO. It is documented.
<invoke partnerLink="scheduling"
    portType="lns:schedulingPT"
    operation="requestProductionScheduling"
    inputVariable="PO">
    <documentation>
        Initiate Production Scheduling
    </documentation>
</invoke>
    In BPELs, it is operator invoke:
// Initiate Production Scheduling
invoke (partnerLink scheduling,
    portType lns:schedulingPT,
    operation requestProductionScheduling,
    inputVariable PO);
    This activity invokes the partner link scheduling with operation
sendShippingSchedule of type lns:schedulingPT and delivers input from
variable shippingSchedule. It is documented and it is synchronized as a target
on the link ship-to-scheduling.
<invoke partnerLink="scheduling"
    portType="lns:schedulingPT"
    operation="sendShippingSchedule"
    inputVariable="shippingSchedule">
    <documentation>
        Complete Production Scheduling
    </documentation>
    <targets>
        <target linkName="ship-to-scheduling" />
    </targets>
</invoke>

```

In BPELs, it is operator invoke:

```
// Complete Production Scheduling
invoke (partnerLink scheduling,
  portType lns:schedulingPT,
  operation sendShippingSchedule,
  inputVariable shippingSchedule,
  target ship-to-scheduling);
  Here is the end of third sequence:
</sequence>
  In BPELs:
};
  Here is end of the flow:
</flow>
  In BPELs:
};
  Finally, the last activity is reply on partner link purchasing on operation
sendPurchaseOrder of port type lns:purchaseOrderPT and delivering the
contents of variable Invoice. It is documented.
<reply partnerLink="purchasing"
  portType="lns:purchaseOrderPT"
  operation="sendPurchaseOrder"
  variable="Invoice">
  <documentation>Invoice Processing</documentation>
</reply>
  In BPELs, it is operator reply:
// Invoice Processing
reply (partnerLink purchasing,
  portType lns:purchaseOrderPT,
  operation sendPurchaseOrder,
  variable Invoice);
  Here is the end of top level activity:
</sequence>
  In BPELs:
};
  Here is end of the process:
</process>
  In BPELs:
};
  In this example, many constructs have been illustrated: sequence, flow, fault
handler, reply, invoke etc. The full examples code is give in the appendix.

```

4 Conclusion

In this paper, in two examples we have shown the main concept of BPELs. This language is really simpler that WS-BPEL and can be used for research and teaching purposes. The next step is to present its semantics via CSP.

Acknowledgments. This research is supported by Project 240/2010 “Development of Grid infrastructure for research and education” funded by the Scientific Research Fund of University of Sofia.

References

1. WS-BPEL, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
2. Service-Oriented Architecture, http://en.wikipedia.org/wiki/Service-oriented_architecture
3. ter Beek, M.; Bucchiarone, A.; Gnesi, S. A Survey on Service Composition Approaches: From Industrial Standards to Formal Methods; Technical report, 2006-15.
4. Communicating Sequential Processes, http://en.wikipedia.org/wiki/Communicating_sequential_processes

Appendix: WS-BPEL Example Code

WSDL definitions used in the example:

```
<wsdl:definitions
  targetNamespace=
    "http://manufacturing.org/wsdl/purchase"
  xmlns:sns="http://manufacturing.org/xsd/purchase"
  xmlns:pos="http://manufacturing.org/wsdl/purchase"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:plnk=
    "http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema>
      <xsd:import
        namespace="http://manufacturing.org/xsd/purchase"
        schemaLocation=
          "http://manufacturing.org/xsd/purchase.xsd" />
      </xsd:schema>
    </wsdl:types>
  <wsdl:message name="POMessage">
    <wsdl:part name="customerInfo"
      type="sns:customerInfoType" />
    <wsdl:part name="purchaseOrder"
      type="sns:purchaseOrderType" />
  </wsdl:message>
  <wsdl:message name="InvMessage">
    <wsdl:part name="IVC" type="sns:InvoiceType" />
  </wsdl:message>
  <wsdl:message name="orderFaultType">
    <wsdl:part name="problemInfo"
      element="sns:OrderFault" />
  </wsdl:message>
```

```

</wsdl:message>
<wsdl:message name="shippingRequestMessage">
  <wsdl:part name="customerInfo"
    element="sns:customerInfo" />
</wsdl:message>
<wsdl:message name="shippingInfoMessage">
  <wsdl:part name="shippingInfo"
    element="sns:shippingInfo" />
</wsdl:message>
<wsdl:message name="scheduleMessage">
  <wsdl:part name="schedule"
    element="sns:scheduleInfo" />
</wsdl:message>
<!-- portTypes supported by the purchase order
process -->
<wsdl:portType name="purchaseOrderPT">
  <wsdl:operation name="sendPurchaseOrder">
    <wsdl:input message="pos:POMessage" />
    <wsdl:output message="pos:InvMessage" />
    <wsdl:fault name="cannotCompleteOrder"
      message="pos:orderFaultType" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="invoiceCallbackPT">
  <wsdl:operation name="sendInvoice">
    <wsdl:input message="pos:InvMessage" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="shippingCallbackPT">
  <wsdl:operation name="sendSchedule">
    <wsdl:input message="pos:scheduleMessage" />
  </wsdl:operation>
</wsdl:portType>
<!-- portType supported by the invoice services -->
<wsdl:portType name="computePricePT">
  <wsdl:operation name="initiatePriceCalculation">
    <wsdl:input message="pos:POMessage" />
  </wsdl:operation>
  <wsdl:operation name="sendShippingPrice">
    <wsdl:input message="pos:shippingInfoMessage" />
  </wsdl:operation>
</wsdl:portType>
<!-- portType supported by the shipping service -->
<wsdl:portType name="shippingPT">
  <wsdl:operation name="requestShipping">
    <wsdl:input message="pos:shippingRequestMessage" />
    <wsdl:output message="pos:shippingInfoMessage" />
    <wsdl:fault name="cannotCompleteOrder"
      message="pos:orderFaultType" />
  </wsdl:operation>
</wsdl:portType>
<!-- portType supported by the production
scheduling process -->
<wsdl:portType name="schedulingPT">

```

```

    <wsdl:operation name="requestProductionScheduling">
      <wsdl:input message="pos:POMessage" />
    </wsdl:operation>
    <wsdl:operation name="sendShippingSchedule">
      <wsdl:input message="pos:scheduleMessage" />
    </wsdl:operation>
  </wsdl:portType>
  <plnk:partnerLinkType name="purchasingLT">
    <plnk:role name="purchaseService"
      portType="pos:purchaseOrderPT" />
  </plnk:partnerLinkType>
  <plnk:partnerLinkType name="invoicingLT">
    <plnk:role name="invoiceService"
      portType="pos:computePricePT" />
    <plnk:role name="invoiceRequester"
      portType="pos:invoiceCallbackPT" />
  </plnk:partnerLinkType>
  <plnk:partnerLinkType name="shippingLT">
    <plnk:role name="shippingService"
      portType="pos:shippingPT" />
    <plnk:role name="shippingRequester"
      portType="pos:shippingCallbackPT" />
  </plnk:partnerLinkType>
  <plnk:partnerLinkType name="schedulingLT">
    <plnk:role name="schedulingService"
      portType="pos:schedulingPT" />
  </plnk:partnerLinkType>
</wsdl:definitions>

```

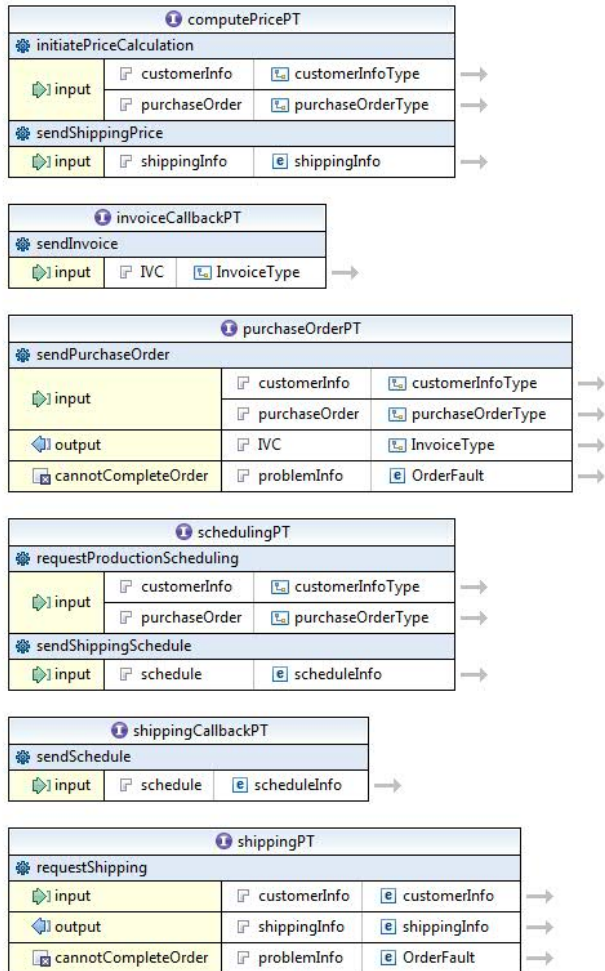



Fig. 1. WSDL example services as IBM Rational Software Architect services diagram.

WS-BPEL code of the example:

```
<process name="purchaseOrderProcess"
targetNamespace="http://example.com/ws-bp/purchase"
xmlns="http://docs.oasis-open.org/wsbpel/2.0/
process/executable"
xmlns:tns="http://manufacturing.org/wsd1/purchase">
<documentation xml:lang="EN">
A simple example of a WS-BPEL process for handling
```

```

    a purchase order.
</documentation>
</partnerLinks>
<partnerLinks>
  <partnerLink name="purchasing"
    partnerLinkType="lns:purchasingLT"
    myRole="purchaseService" />
  <partnerLink name="invoicing"
    partnerLinkType="lns:invoicingLT"
    myRole="invoiceRequester"
    partnerRole="invoiceService" />
  <partnerLink name="shipping"
    partnerLinkType="lns:shippingLT"
    myRole="shippingRequester"
    partnerRole="shippingService" />
  <partnerLink name="scheduling"
    partnerLinkType="lns:schedulingLT"
    partnerRole="schedulingService" />
</partnerLinks>
<variables>
  <variable name="PO" messageType="lns:POMessage" />
  <variable name="Invoice"
    messageType="lns:InvMessage" />
  <variable name="shippingRequest"
    messageType="lns:shippingRequestMessage" />
  <variable name="shippingInfo"
    messageType="lns:shippingInfoMessage" />
  <variable name="shippingSchedule"
    messageType="lns:scheduleMessage" />
</variables>
<faultHandlers>
  <catch faultName="lns:cannotCompleteOrder"
    faultVariable="POFault"
    faultMessageType="lns:orderFaultType">
    <reply partnerLink="purchasing"
      portType="lns:purchaseOrderPT"
      operation="sendPurchaseOrder"
      variable="POFault"
      faultName="cannotCompleteOrder" />
  </catch>
</faultHandlers>
<sequence>
  <receive partnerLink="purchasing"
    portType="lns:purchaseOrderPT"
    operation="sendPurchaseOrder"
    variable="PO"
    createInstance="yes">
    <documentation>
      Receive Purchase Order
    </documentation>
  </receive>
  <flow>
    <documentation>
      A parallel flow to handle shipping, invoicing
      and scheduling

```

```

</documentation>
<links>
  <link name="ship-to-invoice" />
  <link name="ship-to-scheduling" />
</links>
<sequence>
  <assign>
    <copy>
      <from>$PO.customerInfo</from>
      <to>$shippingRequest.customerInfo</to>
    </copy>
  </assign>
  <invoke partnerLink="shipping"
    portType="lns:shippingPT"
    operation="requestShipping"
    inputVariable="shippingRequest"
    outputVariable="shippingInfo">
    <documentation>
      Decide On Shipper
    </documentation>
    <sources>
      <source linkName="ship-to-invoice" />
    </sources>
  </invoke>
  <receive partnerLink="shipping"
    portType="lns:shippingCallbackPT"
    operation="sendSchedule"
    variable="shippingSchedule">
    <documentation>
      Arrange Logistics
    </documentation>
    <sources>
      <source linkName="ship-to-scheduling" />
    </sources>
  </receive>
</sequence>
<sequence>
  <invoke partnerLink="invoicing"
    portType="lns:computePricePT"
    operation="initiatePriceCalculation"
    inputVariable="PO">
    <documentation>
      Initial Price Calculation
    </documentation>
  </invoke>
  <invoke partnerLink="invoicing"
    portType="lns:computePricePT"
    operation="sendShippingPrice"
    inputVariable="shippingInfo">
    <documentation>
      Complete Price Calculation
    </documentation>
  <targets>
    <target linkName="ship-to-invoice" />
  </targets>
</sequence>

```

```

        </targets>
    </invoke>
    <receive partnerLink="invoicing"
        portType="lns:invoiceCallbackPT"
        operation="sendInvoice"
        variable="Invoice" />
</sequence>
<sequence>
    <invoke partnerLink="scheduling"
        portType="lns:schedulingPT"
        operation="requestProductionScheduling"
        inputVariable="PO">
        <documentation>
            Initiate Production Scheduling
        </documentation>
    </invoke>
    <invoke partnerLink="scheduling"
        portType="lns:schedulingPT"
        operation="sendShippingSchedule"
        inputVariable="shippingSchedule">
        <documentation>
            Complete Production Scheduling
        </documentation>
        <targets>
            <target linkName="ship-to-scheduling" />
        </targets>
    </invoke>
</sequence>
</flow>
<reply partnerLink="purchasing"
    portType="lns:purchaseOrderPT"
    operation="sendPurchaseOrder"
    variable="Invoice">
    <documentation>Invoice Processing</documentation>
</reply>
</sequence>
</process>

```

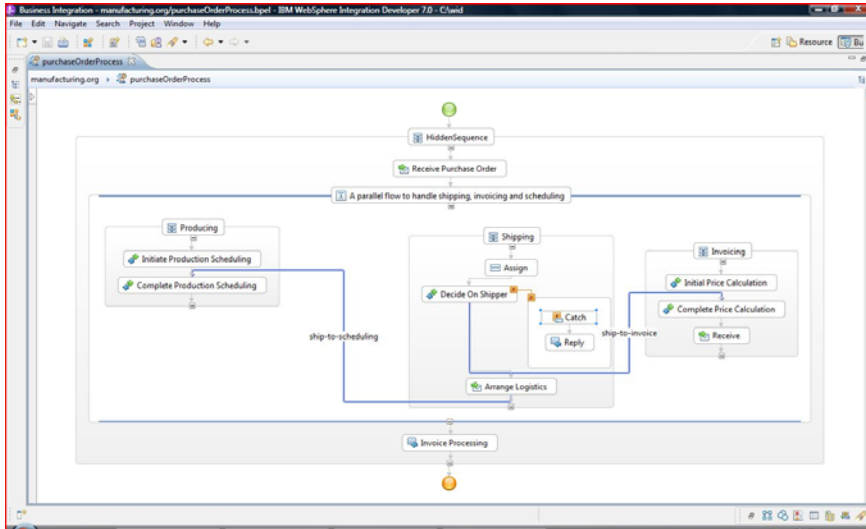


Fig. 2. WS-BPEL example code as IBM WebSphere Integration Developer process diagram.

BPELs example code:

```
// A simple example of a WS-BPEL process
// for handling a purchase order.
process purchaseOrderProcess (
  targetNamespace="http://example.com/ws-bp/purchase",
  xmlns="http://docs.oasis-
    open.org/wsbpel/2.0/process/executable",
  xmlns:lns="http://manufacturing.org/wsd1/purchase") {
  partnerLinks
    lns:purchasingLT purchasing(myRole purchaseService);
    lns:invoicingLT invoicing(myRole invoiceRequester,
      partnerRole invoiceService);
    lns:shippingLT shipping(myRole shippingRequester,
      partnerRole shippingService);
    lns:schedulingLT scheduling(partnerRole
      schedulingService);
  variables
    lns:POMessage PO;
    lns:InvMessage Invoice;
    lns:shippingRequestMessage shippingRequest;
    lns:shippingInfoMessage shippingInfo;
    lns:scheduleMessage shippingSchedule;
  faultHandlers
    catch (faultName lns:cannotCompleteOrder,
      faultVariable lns:orderFaultType POFault) {
      reply (partnerLink purchasing,
        portType lns:purchaseOrderPT,
        operation sendPurchaseOrder,
        variable POFault,
        faultName cannotCompleteOrder);
    }
}
```

```

};
sequence {
  // Receive Purchase Order
  receive (partnerLink purchasing,
    portType lns:purchaseOrderPT,
    operation sendPurchaseOrder,
    variable PO, createInstance yes);
  // A parallel flow to handle shipping,
  // invoicing and scheduling
  flow {
    links ship-to-invoice, ship-to-scheduling;
    sequence {
      $shippingRequest.customerInfo := $PO.customerInfo;
      // Decide On Shipper
      invoke (partnerLink shipping,
        portType lns:shippingPT,
        operation requestShipping,
        inputVariable shippingRequest,
        outputVariable shippingInfo,
        source ship-to-invoice);
      // Arrange Logistics
      receive (partnerLink shipping,
        portType lns:shippingCallbackPT,
        operation sendSchedule,
        variable shippingSchedule,
        source ship-to-scheduling);
    };
    sequence {
      // Initial Price Calculation
      invoke (partnerLink invoicing,
        portType lns:computePricePT,
        operation initiatePriceCalculation,
        inputVariable PO);
      // Complete Price Calculation
      invoke (partnerLink invoicing,
        portType lns:computePricePT,
        operation sendShippingPrice,
        inputVariable shippingInfo,
        target ship-to-invoice);
      receive (partnerLink invoicing,
        portType lns:invoiceCallbackPT,
        operation sendInvoice, variable Invoice);
    };
  };
  sequence {
    // Initiate Production Scheduling
    invoke (partnerLink scheduling,
      portType lns:schedulingPT,
      operation requestProductionScheduling,
      inputVariable PO);
    // Complete Production Scheduling
    invoke (partnerLink scheduling,
      portType lns:schedulingPT,
      operation sendShippingSchedule,
      inputVariable shippingSchedule,

```

```
        target ship-to-scheduling);
    };
};
// Invoice Processing
reply (partnerLink purchasing,
      portType lns:purchaseOrderPT,
      operation sendPurchaseOrder, variable Invoice);
};
};
```

Open MP parallel version of Loop's mesh subdivision algorithm

Peter Armyanov¹

¹ Sofia University, Faculty of Mathematics and Informatics, Sofia, Bulgaria
parmyanov@fmi.uni-sofia.bg

Abstract. This paper presents a parallel version of the well known Loop algorithm for mesh subdivision. The algorithm is split in four major steps, so that each one of them can be easily adapted for parallel execution. All steps are examined and empirical evaluation of the efficiency of their parallel implementation is given. As a result specific parts of the code which can be executed in parallel are selected. A comparison between different objects and core numbers is made, based on the performance of the implementation of the algorithm in Open MP.

Keywords: Mesh subdivision, parallel algorithms, Open MP, Loop's algorithm.

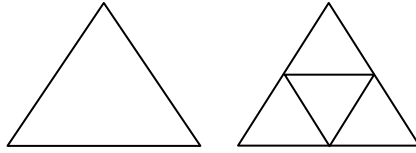
1 Introduction

The creation of smooth objects requires a large number of polygons (sometimes up to a few million). During the elaboration of the model a rough model is created and worked on which later is smoothed automatically with proper algorithms. Also, when transferring 3D objects on a network, smooth objects containing much more information are not expedient concerning the speed of transfer. Therefore smoothing algorithms occupy an important place in the world of 3D computer graphics. One of the most popular and frequently used smoothing algorithm is Loop's algorithm [1]. A tendency in algorithms development is achieving higher performance through realization which works in parallel on several processors or cores. The current article examines a possible approach to a parallel implementation of Loop's mesh subdivision algorithm, using OpenMP.

2 Description of Algorithm

Loop's algorithm is based on the addition of new vertices in the 3D object, so that the newly created model represents its smoother version. Each triangle of the primary mesh is divided into four new triangles by adding a vertex on each of its edges.





Next, each of the vertices on the new object is moved by an offset computed with a formula which gives the new coordinates depending on the current coordinates and the number of vertices the current vertex is connected with. This formula is based on B-Splines and their properties of achieving smoother forms.

3 Parallel Implementation.

The suggested parallel implementation of Loop's algorithm takes advantage of its simplicity and its ability to be divided into several consecutive steps, which can be easily executed in parallel.

The program processes the object by firstly storing the coordinates of all its vertices, edges and triangles. Additional information is stored for each vertex concerning which points it is connected to. Edges require tracking information about the triangles they are part of, as well as the indexes of the vertices they connect. The triangles themselves store information about the edges they are composed of and their normal vectors. This information is required for proper object visualization.

The implemented algorithm consists of the following four steps:

1. a vertex is added in the middle of each edge of the object;
2. four new triangles are created from each triangle, based on the added in the previous step vertices;
3. each of the vertices in the object changes its coordinates depending on the vertices it is connected to;

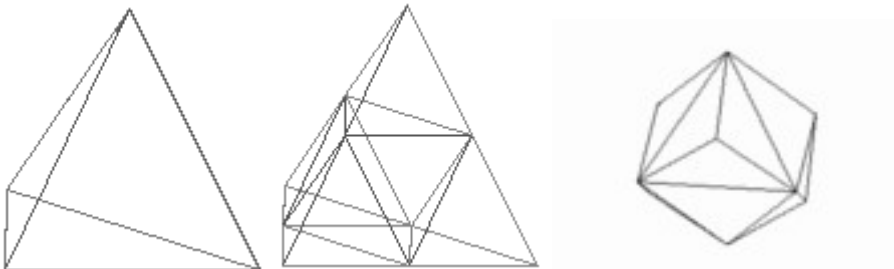


Fig 1. a simple 3D object before applying the algorithm, after the second and third step

The first step is performed independently for each edge of the figure. Therefore it can be easily parallelized. The addition of new vertices is possible without overhead caused by synchronization if each vertex is stored at a given index and in a random access data structure. For this implementation is used STL vector with pre-allocation. The index of each new vertex can be computed from the index of the edge it lies on and the number of the original mesh vertices.

The second step is the most complicated and requires modification of all the information stored for the object. In this step new edges and triangles are added. Parallel execution can be applied for each triangle. For each of them, three new edges are added, the vertices of the original triangle are changed, and three new triangles are added. This step requires synchronization of the parallel processing. An important issue is to avoid processing one edge more than once. Thus it is suitable to add information for each edge on if it was processed or not. Access to this access must be atomic. This synchronization decreases slightly the performance of the program, but cannot be avoided for the reason that each of the edges may be processed as part of exactly two triangles. Again, by using proper index access for storing triangles and edges no synchronization is needed.

The third step is implemented using the vertices - again in parallel. As a result an array of new vertices is created. This is necessary because for every vertex the new coordinates are calculated on the basis of the original coordinates of the vertices it was previously connected with. To optimize this procedure the suggested implementation stored not only the information about the number of vertices connected to each vertex, but also their indexes. This information is easy to maintain and synchronize. After calculating the new coordinates of all the vertices and their storage in an array the invalid information is overwritten with the new data.

In the last step the normal vector is calculated for each triangle. This is an operation which can be performed in parallel for each triangle.

Thus, implementing the well known Loop algorithm as a sequence of four steps, each of which may be processed in parallel using OpenMP, a new version compatible with the contemporary multi-core and multiprocessor architectures is achieved.

4 Empirical Results and Analyze.

It is well known that the usage of methods for parallel execution of parts of the code may lead to slow down in comparison with the execution of the same part of the code without parallel execution [2]. After a series of empirical tests it was established that the biggest efficiency in parallel implementation takes part during the second and third steps of the algorithm. It was established also that for the first step a parallel implementation is not as appropriate because adding the vertices is an operation with high efficiency and fast. In this step even for over one million triangles, the parallel implementation is not as efficient as the consecutive one. For the other parts of the algorithm the parallel implementation increases the performance for objects with over 4000 vertices.

The execution time (in seconds) for the different parts of the consecutive algorithm with objects of different triangles in their meshes can be seen in table 1. The same parameters for parallel version are shown in table 2. When executing on more than one core or processors, the primary overhead of the parallel processing is bigger. Objects with more complex meshes are needed to compensate the overhead of the parallel execution.

Triangles	Step 1	Step 2	Step 3	Step 4
16	0,000	0,000	0,000	0,000
64	0,000	0,000	0,000	0,000
256	0,000	0,010	0,006	0,000
1024	0,000	0,032	0,011	0,000
4096	0,000	0,108	0,025	0,001
16384	0,002	0,270	0,054	0,005
65536	0,006	0,871	0,130	0,018
262144	0,014	3,145	0,283	0,076
1048576	0,022	12,572	0,597	0,331

Table 1. Execution time in seconds for consecutive execution of the different steps in Loop's Algorithm for objects with different triangle numbers in their meshes

Triangles	Step 1	Step 2	Step 3	Step 4
16	0,000	0,000	0,000	0,000
64	0,001	0,000	0,000	0,000
256	0,003	0,008	0,009	0,000
1024	0,004	0,012	0,010	0,001
4096	0,006	0,042	0,014	0,002
16384	0,010	0,170	0,028	0,002
65536	0,012	0,516	0,061	0,011
262144	0,019	2,021	0,124	0,038
1048576	0,042	8,012	0,314	0,162

Table 2. Execution time in seconds for parallel execution of the different steps in Loop's Algorithm for objects with different triangle numbers in their meshes

Table 3 shows the time (in seconds) for smoothing objects with different vertex number in the consecutive implementation of the algorithm, as well as the time for execution in 2, 4 and 16 core processor.

Triangles	sequential	2 cores	4 cores	16 cores
16	0,000	0,000	0,000	0,000
64	0,000	0,000	0,000	0,006
256	0,015	0,016	0,018	0,022
1024	0,032	0,015	0,030	0,050
4096	0,094	0,063	0,060	0,080
16384	0,312	0,188	0,093	0,010
65536	0,953	0,563	0,291	0,090
262144	3,484	1,953	1,213	0,300
1048576	13,563	8,313	4,710	1,100

Table 3. Time in seconds for execution of Loop's algorithm for objects with different triangle numbers on 1, 2, 4 and 16 core processor

It can be easily seen that for a small number of triangles the parallel implementation gives slower results than the consecutive one. Also, the more cores take part in the computation process for objects with small triangle mesh, the slower

the results are. In large triangle number (over 50 000) the overhead of synchronization and control process is compensated by the acceleration of the parallel work. For complex objects the parallel version of the algorithm is significantly faster.

3 Conclusion and Future Work

The result of the made modifications in the classical form of Loop's Algorithm is its released version which was easily divided into parts for implementation in parallel using OpenMP as a technology for parallel programming. The analysis of the parts of the code which are suitable for parallel implementation made in this paper prove that acceleration of the performance can be made proportional to the number of processor cores on which the program runs. The parallel version of the algorithm can be used for optimization of transfer of 3D objects by decreasing the number of polygons and relying on later subdivision.

One drawback of the current algorithm is that it can be applied only on objects with triangular meshes. A future development of the algorithm will eliminate this problem by triangulating any non-triangular mesh as a primary step..

5 References

1. Charles T. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.
2. OpenMP C/C++ Application Programming Interface, March 2002. Version 2.0.
3. Stephan Bischoff. Towards Hardware Implementation Of Loop Subdivision. SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware, Interlaken, Switzerland, 2000
4. Tien-hsiung Weng, Barbara Chapman. Towards optimisation of openMP codes for synchronisation and data reuse. International Journal of High Performance Computing and Networking. Volume 1. 2004
5. Empirical Studies of Programmers, Proceedings usually published by ACM Press. 1993, 1996, 1997.

Intelligent Systems

Comparison of ACO behavior with various start strategies applied on MKP

Stefka Fidanova¹, Pencho Marinov¹ and Krassimir Atanassov²

¹IPP – Bulgarian Academy of Sciences, Acad. G. Bonchev str. bl.25A, 1113 Sofia, Bulgaria, {stefka,pencho}@parallel.bas.bg

²CLBME – Bulgarian Academy of Science, Acad. G. Bonchev str, bl 105, 1113 Sofia, Bulgaria, krat@bas.bg

Abstract. Ant Colony Optimization is a stochastic search method that mimic the social behavior of real ants colonies, which manage to establish the shortest routs to feeding sources and back. Such algorithms have been developed to arrive at near-optimum solutions to large-scale optimization problems, for which traditional mathematical techniques may fail. In this paper on each iteration estimations of the start nodes of the ants are made. Two sets of start strategies are prepared. Benchmark comparisons among them are presented in terms of quality of the results. Based on this comparison analysis, the performance of the algorithm is discussed along with some guidelines for determining the best strategy. The study presents ideas that should be beneficial to both practitioners and researchers involved in solving optimization problems.

1 Introduction

The difficulties associated with using mathematical optimization on large-scale engineering problems, have contributed to the development of alternative solutions. Linear programming and dynamic programming techniques, for example, often fail in solving NP-hard problems with large number of variables. To overcome these problems, researchers have proposed metaheuristic methods for searching near-optimal solutions to problems. One of the most successful metaheuristic is Ant Colony Optimization (ACO).

Real ants foraging for food lay down quantities of pheromone (chemical cues) marking the path that they follow. An isolated ant moves essentially at random but an ant encountering a previously laid pheromone will detect it and decide to follow it with high probability and thereby reinforce it with a further quantity of pheromone. The repetition of the above mechanism represents the auto-catalytic behavior of a real ant colony where the more the ants follow a trail, the more attractive that trail becomes.

ACO is inspired by real ant behavior to solve hard combinatorial optimization problems. Examples of hard optimization problems are Traveling Salesman Problem [9], Vehicle Routing [10], Minimum Spanning Tree [7], Constrain Satisfaction [5], Knapsack Problem [3], etc. The ACO algorithm uses a colony of artificial ants that behave as cooperative agents in a mathematical space where



they are allowed to search and reinforce pathways (solutions) in order to find the optimal ones. The problem is represented by graph and the ants walk on the graph to construct solutions. The solutions are represented by paths in the graph. After the initialization of the pheromone trails, the ants construct feasible solutions, starting from random nodes, and then the pheromone trails are updated. At each step the ants compute a set of feasible moves and select the best one (according to some probabilistic rules) to continue the rest of the tour. The structure of the ACO algorithm is shown by the pseudocode below. The transition probability $p_{i,j}$, to choose the node j when the current node is i , is based on the heuristic information $\eta_{i,j}$ and the pheromone trail level $\tau_{i,j}$ of the move, where $i, j = 1, \dots, n$.

$$p_{i,j} = \frac{\tau_{i,j}^a \eta_{i,j}^b}{\sum_{k \in Unused} \tau_{i,k}^a \eta_{i,k}^b}, \quad (1)$$

where *Unused* is the set of unused nodes of the graph.

The higher the value of the pheromone and the heuristic information, the more profitable it is to select this move and resume the search. In the beginning, the initial pheromone level is set to a small positive constant value τ_0 ; later, the ants update this value after completing the construction stage. ACO algorithms adopt different criteria to update the pheromone level.

Ant Colony Optimization

```

Initialize number of ants;
Initialize the ACO parameters;
while not end-condition do
    for k=0 to number of ants
        ant k choses start node;
        while solution is not constructed do
            ant k selects higher probability node;
        end while
    end for
    Update-pheromone-trails;
end while

```

Fig. 1. Pseudocode for ACO

The pheromone trail update rule is given by:

$$\tau_{i,j} \leftarrow \rho \tau_{i,j} + \Delta \tau_{i,j}, \quad (2)$$

where ρ models evaporation in the nature and $\Delta \tau_{i,j}$ is new added pheromone which is proportional to the quality of the solution.

The novelty in this work is the use of estimations of start nodes with respect to the quality of the solution and thus to better manage the search process.

Two sets of start strategies and their combinations are offered. Like test is used Multiple Knapsack Problem (MKP) as a representative of subset problems.

The rest of the paper is organized as follows. In section 2 estimation of start regions and two sets of start strategies are proposed. In section 3 the new ideas are applied on MKP and computational results are analyzed. At the end some conclusions and directions for future work are done.

2 Start Strategies

The known ACO algorithms create a solution starting from random node. But for some problems, especially subset problems, it is important from which node the search process starts. For example if an ant starts from node which does not belong to the optimal solution, probability to construct it is zero. In this paper is proposed two sets of start strategies. The aim is to use the experience of the ants from previous iteration to choose the better starting node. Other authors use this experience only by the pheromone, when the ants construct the solutions.

Let the graph of the problem has m nodes. The set of nodes is divided on N subsets. There are different ways for dividing. Normally, the nodes of the graph are randomly enumerated. An example for creating of the subsets, without lost of generality, is: the node number one is in the first subset, the node number two - in the second subset, etc., the node number N is in the N -th subset, the node number $N + 1$ is in the first subset, etc. Thus the number of nodes in the separate subsets are almost equal. After the first iteration the estimations $D_j(i)$ and $E_j(i)$ are introduced of the node subsets, where $i \geq 2$ is the number of the current iteration and $D_j(i)$ and $E_j(i)$ are weight coefficients of j -th node subset ($1 \leq j \leq N$), which are calculated by the following formulas:

$$D_j(i) = \frac{i \cdot D_j(i-1) + F_j(i)}{i}, \quad (3)$$

$$E_j(i) = \frac{i \cdot E_j(i-1) + G_j(i)}{i}, \quad (4)$$

where $i \geq 2$ is the current iteration and for each j ($1 \leq j \leq N$):

$$F_j(i) = \begin{cases} \frac{f_{j,A}}{n_j} & \text{if } n_j \neq 0 \\ F_j(i-1) & \text{otherwise} \end{cases}, \quad (5)$$

$$G_j(i) = \begin{cases} \frac{g_{j,B}}{n_j} & \text{if } n_j \neq 0 \\ G_j(i-1) & \text{otherwise} \end{cases}, \quad (6)$$

and $f_{j,A}$ is the number of the solutions among the best $A\%$, and $g_{j,B}$ is the number of the solutions among the worst $B\%$, where $A + B \leq 100$, $i \geq 2$ and $\sum_{j=1}^N n_j = n$, where n_j ($1 \leq j \leq N$) is the number of solutions obtained by

ants starting from nodes subset j . Initial values of the weight coefficients are : $D_j(1) = 1$ and $E_j(1) = 0$.

Let threshold E for $E_j(i)$ and D for $D_j(i)$ be fixed, than two sets of strategies to choose start node for every ant are constructed, the threshold E increase every iteration with $1/i$ where i is the number of the current iteration:

Start Strategies I

- 1 If $E_j(i) > E$ then the subset j is forbidden for current iteration and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 2 If $E_j(i) > E$ then the subset j is forbidden for current simulation and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 3 If $E_j(i) > E$ then the subset j is forbidden for K_1 consecutive iterations and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 4 Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is chosen randomly from subset $\{j \mid D_j(i) > D\}$, otherwise a node is chosen randomly from the not forbidden subsets, R is chosen and fixed at the beginning.
- 5 Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is randomly chosen from subset $\{j \mid D_j(i) > D\}$, otherwise a node is randomly chosen from the not forbidden subsets, R is chosen at the beginning and increase with r_3 every iteration.

Start Strategies II

- 1 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for current iteration and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 2 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for current simulation and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 3 If $E_j(i)/D_j(i) > E$ then the subset j is forbidden for K_1 consecutive iterations and the starting node is chosen randomly from $\{j \mid j \text{ is not forbidden}\}$;
- 4 Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is chosen randomly from subset $\{j \mid D_j(i) > D\}$, otherwise a node is chosen randomly from the not forbidden subsets, R is chosen and fixed at the beginning.
- 5 Let $r_1 \in [R, 1)$ is a random number. Let $r_2 \in [0, 1]$ is a random number. If $r_2 > r_1$ a node is randomly chosen from subset $\{j \mid D_j(i) > D\}$, otherwise a node is randomly chosen from the not forbidden subsets, R is chosen at the beginning and increase with r_3 every iteration.

Where $0 \leq K_1 \leq$ "number of iterations" is a parameter. If $K_1 = 0$, than strategy 3 is equal to the random choose of the start node. If $K_1 = 1$, than strategy 3 is equal to the strategy 1. If $K_1 =$ "maximal number of iterations", than strategy 3 is equal to the strategy 2. The strategies 1, 2 and 3 can be called forbid strategies, and strategies 4 and 5 can be called stimulate strategies. By stimulate strategies the ants are forced to start there search from subsets with high value of $D_j(i)$. If $R = 0.5$, than the probability an ant to start from nodes subset with high value of $D_j(i)$ is two times high than to start from other subset.

The difference between the sets of strategies is in the strategies 1, 2 and 3. In strategy set I to forbid some region is used only the estimation $E_j(i)$ how bad is this region and in the strategy subset II is used $E_j(i)/D_j(i)$ and thus is taken in to account how many bad and how many good solutions start from the same node subset. The aim is balance between the bad and good estimation for every region.

3 Experimental Results

The start strategy performance is analyzed in this section. Like test is used Multiple Knapsack Problem (MKP) because it is subset problem. The Multiple Knapsack Problem has numerous applications in theory as well as in practice. It also arise as a subproblem in several algorithms for more complex problems and these algorithms will benefit from any improvement in the field of MKP. The following major applications can be mentioned: problems in cargo loading, cutting stock, bin-packing, budget control and financial management may be formulated as MKP. In [8] is proposed to use the MKP in fault tolerance problem and in [1] is designed a public cryptography scheme whose security realize on the difficulty of solving the MKP. In [6] is mentioned that two-processor scheduling problems may be solved as a MKP. Other applications are industrial management, naval, aerospace, computational complexity theory.

The MKP can be thought as a resource allocation problem, where there are m resources (the knapsacks) and n objects and every object j has a profit p_j . Each resource has its own budget c_j (knapsack capacity) and consumption r_{ij} of resource i by object j . The aim is maximizing the sum of the profits, while working with a limited budget.

The MKP can be formulated as follows:

$$\begin{aligned} & \max \sum_{j=1}^n p_j x_j \\ & \text{subject to } \sum_{j=1}^n r_{ij} x_j \leq c_i \quad i = 1, \dots, m \\ & x_j \in \{0, 1\} \quad j = 1, \dots, n \end{aligned} \tag{7}$$

x_j is 1 if the object j is chosen and 0 otherwise.

There are m constraints in this problem, so MKP is also called m -dimensional knapsack problem. Let $I = \{1, \dots, m\}$ and $J = \{1, \dots, n\}$, with $c_i \geq 0$ for all $i \in I$. A well-stated MKP assumes that $p_j > 0$ and $r_{ij} \leq c_i \leq \sum_{j=1}^n r_{ij}$ for all $i \in I$ and $j \in J$. Note that the $[r_{ij}]_{m \times n}$ matrix and $[c_i]_m$ vector are both non-negative.

In the MKP one is not interested in solutions giving a particular order. Therefore a partial solution is represented by $S = \{i_1, i_2, \dots, i_j\}$ and the most recent elements incorporated to S , i_j need not be involved in the process for selecting the next element. Moreover, solutions for ordering problems have a fixed length as one search for a permutation of a known number of elements. Solutions for MKP, however, do not have a fixed length. The graph of the problem

is defined as follows: the nodes correspond to the items, the arcs fully connect nodes. Fully connected graph means that after the object i one can choose the object j for every i and j if there are enough resources and object j is not chosen yet.

The computational experience of the ACO algorithm is shown using 10 MKP instances from “OR-Library” available within WWW access at <http://people.brunel.ac.uk/~mastjjb/jeb/orlib>, with 100 objects and 10 constraints. To provide a fair comparison for the above implemented ACO algorithm, a predefined number of iterations, $k = 100$, is fixed for all the runs. The developed technique has been coded in C++ language and implemented on a Pentium 4 (2.8 Ghz). The parameters are fixed as follows: $\rho = 0.5$, $a = 1$, $b = 1$, number of used ants is 20, $A = 30$, $B = 30$, $D = 1.5$, $E = 0.5$, $K_1 = 5$, $R = 0.5$, $r_3 = 0.01$. The values of ACO parameters (ρ, a, b) are from [4] and experimentally is found that they are best for MKP. The tests are run with 1, 2, 4, 5 and 10 nodes within the nodes subsets. For every experiment, the results are obtained by performing 30 independent runs, then averaging the fitness values obtained in order to ensure statistical confidence of the observed difference. The computational time which takes start strategies is negligible with respect to running time of the algorithm.

Tests with all combinations of strategies and with random start (12 combinations) are run. Thus we perform 18 000 tests for every strategy set. One can observe that sometimes all nodes subsets become forbidden and the algorithm stops before performing all iterations (strategies 1, 2, 3 and combinations with them). In this situation there are two possibilities. The first is to report the achieved result when the algorithm stops. The second possibility is to continue the algorithm, applying random start for several iteration till some of the subsets become not forbidden. The second possibility improves the achieved results with respect to the first one.

Average achieved result by some strategy, is better than without any strategy, for every test problem. For fair comparison, the difference d between the worst and best average result for every problem is divided to 10. If the average result for some strategy is between the worst average result and worst average plus $d/10$ it is appreciated with 1. If it is between the worst average plus $d/10$ and worst average plus $2d/10$ it is appreciated with 2 and so on. If it is between the best average minus $d/10$ and the best average, it is appreciated with 10. Thus for a test problem the achieved results for every strategy and every nodes deviation is appreciated from 1 to 10. After that is summed the rate of all test problems for every strategy and every nodes deviation. So their rate becomes between 10 and 100 (see Table 1 and Table 2). It is histogram like representation.

Regarding the strategies (rows) is observed that for most of them the highest rate is when the nodes subsets consists of 1 or 2 nodes. When the nodes subsets consist of 10 nodes the rate is low. The highest rate have strategy combinations between one forbidden and one stimulated strategies. Comparing the sets of strategies we observe that the strategy set II gives better results than strategy set I, when there are more nodes in the node subsets (4, 5, 10). It is because in

Table 1. Estimation of strategies and nodes deviation for set strategy I

number nodes	10	5	4	2	1
random	22	22	22	22	22
strat. 1	23	39	49	82	82
strat. 2	23	39	49	82	82
strat. 3	23	39	49	82	82
strat. 4	71	77	71	78	82
strat. 5	73	76	75	73	80
strat. 1-4	21	45	56	79	87
strat. 1-5	24	40	49	83	84
strat. 2-4	21	45	56	79	87
strat. 2-5	24	40	49	83	84
strat. 3-4	21	45	56	79	87
strat. 3-5	24	40	49	83	84

Table 2. Estimation of strategies and nodes deviation for set strategy I

number nodes	10	5	4	2	1
random	22	22	22	22	22
strat. 1	55	79	79	81	80
strat. 2	52	76	73	81	84
strat. 3	54	80	74	81	80
strat. 4	71	77	71	78	82
strat. 5	74	76	78	81	80
strat. 1-4	55	80	77	85	85
strat. 1-5	52	81	73	85	80
strat. 2-4	54	81	75	84	84
strat. 2-5	50	76	75	85	83
strat. 3-4	54	80	75	85	85
strat. 3-5	55	75	71	86	85

set strategy II there is better balance between estimation for nodes subset to be good and to be bad and when there is many nodes in the node subsets a lot of good and a lot of bad solutions can be started from the same subset. When the node subsets consist 1 or 2 nodes from them starts preferably bad or preferably good solutions, therefore the set strategy I achieves better results in this case.

4 Conclusion

This paper is addressed to ant colony optimization algorithm with controlled start combining two sets of start strategies. So, the start node of each ant depends of the goodness of the respective region. We have analyzed the behavior of the ACO algorithm with different strategies. The achieved solutions with strategies are better than random start. The future work will be focused on parameter settings which manage the starting procedure. It will be investigated on influence of the parameters to algorithm performance. The aim is to study in detail the relationships between the start nodes and the quality of the achieved solutions.

Acknowledgments: This work has been partially supported by the Bulgarian National Scientific Fund under the grants "Modeling Processes with fixed development rules" DID 02/29 and "Effective Monte Carlo Methods for large-scale scientific problems" DTK 02/44.

References

1. Diffe, W., Hellman, M. E., (1976). New direction in cryptography. IEEE Trans Inf. Theory. IT-36, 644-654.
2. Dorigo, M., Gambardella, L.M., (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1,53-66.
3. Fidanova, S., (2002). Evolutionary algorithm for multiple knapsack problem, Int. Conference Parallel Problems Solving from Nature, Real World Optimization Using Evolutionary Computing, ISBN No 0-9543481-0-9,Granada, Spain.
4. Fidanova, S., (2006). Ant colony optimization and multiple knapsack problem, in: Renard, J.Ph. (Eds.), Handbook of Research on Nature Inspired Computing for Economics ad Management, Idea Group Inc., ISBN 1-59140-984-5, 498-509.
5. Lessing, L., Dumitrescu, I., Stutzle, T., (2004), A comparison between ACO algorithms for the set covering problem, ANTS Workshop, 1-12.
6. Martello, S., Toth, P., (1984). A mixtures of dynamic programming and branch-and-bound for the subset-sum problem, Management Science 30, 756-771.
7. Reiman, M., Laumanns, M., (2004). A hybrid ACO algorithm for the capacitate minimum spanning tree problem, Workshop on Hybrid Metahuristics, Valencia, Spain, 1-10.
8. Sinha, A., Zoltner, A. A., (1979). The multiple-choice knapsack problem, J. Operational Research 27, 503-515.
9. Stutzle, T., Dorigo, M., (1999). ACO algorithm for the traveling salesman problem, In Miettinen, K., Makela,M., Neittaanmaki, P., Periaux J. (Eds.), Evolutionary Algorithms in Engineering and Computer Science, Wiley, 163-183.

10. Zhang, T., Wang, S., Tian, W., Zhang, Y., (2006). ACO-VRPTWRV: A new algorithm for the vehicle routing problems with time windows and re-used vehicles based on ant colony optimization, Conference on Intelligent Systems Design and Applications, IEEE press, 390-395.

Providing and Maintaining Interoperability in Digital Library Systems

Maria M. Nisheva-Pavlova¹

¹ Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”,
5 James Bouchier Blvd., Sofia 1164, Bulgaria
marian@fmi.uni-sofia.bg

Abstract. The paper discusses the main characteristics of semantic interoperability in digital library systems. Some widespread standards and open questions in the field of building descriptions of objects and repositories as well as some proper solutions of the variety of problems concerning the development of adequate mechanisms for semantics oriented search in multilingual academic digital libraries are analyzed in brief. An ontology-based approach is suggested in order to standardize the semantic annotation of the library resources and to facilitate the implementation of the functionality of the search engine. This approach is discussed in the context of an ongoing project aimed at the development of a methodology and proper software tools for building academic digital libraries. Some suitable examples are given as illustrations of the suggested ideas and tools.

Keywords: Digital Library, Interoperability, Metadata, Semantic Annotation, Ontology, Semantic Web

1 Introduction

Ever since digital libraries came into view, *interoperability* is becoming a substantial issue in their design and implementation. Interoperability in general is concerned with the capability of differing information systems to communicate. This communication may take various forms such as the transfer, exchange, transformation, mediation, migration or integration of information. If two or more systems are capable of communicating and exchanging data, they are displaying *syntactic interoperability*. Beyond the ability of two (or more) computer systems to exchange information, *semantic interoperability* is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of both systems.

The main focus of this paper is *semantic interoperability in digital library systems*, the goal of which is to facilitate complex and more advanced, context-sensitive query processing over heterogeneous information resources [1].



2 Main Characteristics of Semantic Interoperability

Semantic interoperability is characterised by the capability of different information systems to communicate information consistent with the intended meaning of the encoded information (as intended by the creators or maintainers of the information system). It involves:

- the processing of the shared information so that it is consistent with the intended meaning;
- the encoding of queries and presentation of information so that it conforms with the intended meaning regardless of the source of information.

The DELOS Network of Excellence (<http://www.delos.info/>) defines a 10-year vision for digital libraries: “digital libraries should enable any citizen to access all human knowledge any time and anywhere, in a friendly, multi-modal, efficient and effective way, by overcoming barriers of distance, language, and culture and by using multiple Internet-connected devices”. Semantic interoperability is an essential technology in realizing this goal.

The main areas in which semantic interoperability is important in digital library systems include [1]:

- improving the precision of search;
- enabling advanced search;
- facilitating reasoning over document collections and knowledge bases;
- integration of heterogeneous resources and its relevance in the information life-cycle management process.

Semantic interoperability issues are relevant to different degree in each of the following elements of information lifecycle in which creators/authors, publishers, information systems managers, service providers and/or end-users are involved [1]:

- creation, modification;
- publication;
- acquisition, selection, storage, system and collection building;
- cataloguing (metadata, identification/naming, registration), indexing, knowledge organisation, knowledge representation, modeling;
- integration, brokering, linking, syntactic and semantic interoperability engineering;
- mediation (user interfaces, personalisation, reference, recommendation, transfer etc.);
- access, search and discovery;
- use, shared application/collaboration, scholarly communication, annotation, evaluation, reuse, work environments;
- maintenance;
- archiving and preservation.

Standardization and ontologies are indicated as most effective instruments for providing and maintaining interoperability in information systems and especially in digital library systems.

3 Standardization

Standardization in order to achieve semantic interoperability in a digital library environment may comprise the form and meaning of metadata and content schemata as well as the use of names and construction of identifiers for concepts and real world items.

Standardization may direct to the following favourable features of the respective information (in particular, digital library) systems:

- information can be immediately communicated (transferred, integrated, merged etc.) without transformation;
- information can be communicated without alteration;
- information can be kept in a single form;
- information of candidate sources can be enforced to be functionally complete for an envisaged integrated service.

Many different metadata schemes are being developed as standards across disciplines, such as library science, education, archiving, e-commerce, etc. As most popular metadata standards one should mention:

- Dublin Core (<http://dublincore.org/>) – an interoperable online metadata standard focused on networked resources;
- Encoded Archival Description (EAD, <http://www.loc.gov/ead/>) – a standard for encoding archival finding aids using XML in archival and manuscript repositories;
- Online Information Exchange (ONIX, <http://www.editeur.org/15/Previous-Releases/>) – an international standard for representing and communicating book industry product information in electronic form;
- Learning Objects Metadata (IEEE LOM, <http://ltsc.ieee.org/wg12/>) – a standard which specifies the syntax and semantics of Learning Object Metadata;
- Text Encoding Initiative (TEI, <http://www.tei-c.org/index.xml>) – a standard for the representation of texts in digital form;
- Z39.87 Data Dictionary (NISO MIX, <http://www.loc.gov/standards/mix/>) – a technical metadata standard for a set of technical data elements required to manage digital image collections;
- Machine Readable Cataloging (MARC, <http://www.loc.gov/marc/>) – a set of standards for the representation and communication of bibliographic and related information in machine-readable form;
- Metadata Encoding and Transmission Standard (METS, <http://www.loc.gov/standards/mets/>) – an XML schema for encoding descriptive, administrative, and structural metadata regarding objects within a digital library;
- Multimedia Content Description Interface (MPEG-7, <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>) – a ISO/IEC standard which specifies a set of descriptors to describe various types of multimedia information;
- Directory Interchange Format (DIF, <http://gcmd.gsfc.nasa.gov/User/difguide/difman.html>) – a descriptive and standardized format for exchanging information about scientific data sets.

4 Role of Ontologies

One of the well-accepted mechanisms for achieving semantic interoperability is the utilization of ontologies. According to the most popular formal definition, “*An ontology is an explicit specification of a conceptualization*” [2]. Metadata vocabularies and ontologies are considered as ways of providing semantic context in determining the relevance of resources. Ontologies are usually developed in order to define the meaning of concepts and terms used in a specific domain. The choosing and sharing of vocabulary elements coherently and consistently across applications is known as *ontological commitment* [3] and is a good basis for semantic interoperability in independent and disparate systems.

According to the most popular classification scheme, ontologies can be classified into three main categories:

- *Upper Ontologies*, where basic, domain-independent concepts as well as relationships among them are defined;
- *Core (or Intermediate) Ontologies*, which are comprised of concepts and relationships that are thought to be basic in the broad application domain context. They often capture the semantics of well-accepted domain standards;
- *Domain Ontologies*, where concepts and relationships used in specific application domains are defined. The concepts defined in Domain Ontologies specialise the ones defined in both Upper and Core Ontologies, thus extending them with domain knowledge.

In general, semantic interoperability depends mainly on the existence of well-formed and accepted Upper and Core Ontologies, in which the basic concepts and relationships are defined. Then, the concepts defined in the Upper and Core Ontologies, should be extended by appropriate Domain Ontologies.

The standards for metadata descriptions usually provide only general-purpose structures. Therefore, the utilization of Core and/or Upper Ontologies in capturing the semantics of the standards in combination with Domain Ontologies that extend them with domain knowledge, are systematic mechanisms for the extension and adaptation of standards.

5 An Example: DigLib-CI

5.1 An Overview of DigLib-CI

DigLib-CI is a digital library with freely available research and learning materials (articles, periodicals and single articles, dissertations, monographs, lecture notes, textbooks, presentations, source code of example computer programs, data sets, quizzes, manuals, etc.) in the fields of Computer Science and Information Systems. It has been under development at the Department of Computer Informatics of the Faculty of Mathematics and Informatics (FMI), Sofia University, in order to provide open access to various kinds of scholarly and instructional content. An ontology-based approach is suggested within the respective project in order to standardize the semantic annotation of the library resources and to facilitate the implementation of the

functionality of the search engine. More precisely, this project has the following main objectives:

- to study the various aspects of creation and integration of appropriate ontologies oriented to the contents of multilingual academic digital libraries;
- to define suitable metadata to accompany various types of scholarly and learning materials, taking into account the internationally approved classification schemes and FMI experience;
- to develop a proper framework for application of advanced information technologies and particularly Semantic Web technologies in building tools for semantics oriented search in multilingual digital libraries.

The functional structure of DigLib-CI includes components of six types:

- content repositories;
- metadata catalogues;
- subject ontologies and index;
- a search engine;
- a module implementing the library functionality;
- an interface module.

The content repositories include research and learning materials of different types in the areas of Computer Science and Information Systems, available in various digital formats. The materials in the content repositories are written in Bulgarian or in English language.

The metadata catalogues are destined to facilitate the identification of the needed research or learning materials by the search engine. They contain descriptive metadata and support the reusability of all library resources and facilitate their interoperability.

The subject ontologies include concepts of the Computer Science and Information Systems areas, with description of their properties and different kinds of relationships among them. They play a significant role in the implementation of the full functionality of the search engine.

The purpose of the search engine is to provide adequate access to the complete palette of resources stored in DigLib-CI.

The library functionality and the user interface of DigLib-CI are designed in accordance with the expected requirements of the typical users of the library. The interface module provides adequate online access to the library resources and supporting software tools.

5.2 Metadata, Ontologies and Standards

The library catalogues contain metadata which support the identification of the requested resources by the search engine. These metadata are stored in XML format and comply with the IEEE Standard for Learning Object Metadata [4].

Typical examples of relevant attributes of most kinds of research and learning materials are: type of the material; author; title of the material; language(s) (human and/or programming one(s)); digital format; location; version; date of creation; completion status; restrictions on use; semantic annotation – list of concepts from a proper subject ontology describing the Computer Science or Information Systems subfields and/or concepts treated by the material. The concepts of the subject

ontologies are too general from the viewpoint of the expectations of the users of DigLib-CI, so one can include in the resource descriptions additional lists of keywords which describe the content of the documents at the necessary level of abstraction.

Learning materials have been characterized also by their educational level and the principal types of users for which the corresponding material was designed; officially published research materials and textbooks are supplied with the appropriate bibliographic metadata.

Each catalogue entry (i.e., each resource description) consists of two equivalent parts in which the element values are texts in Bulgarian or English language respectively. The search engine examines the corresponding parts of the descriptions according to the language of the user query.

The subject ontologies include basic concepts in the fields of Computer Science and Information Systems, with description of their properties and the main relationships between them. Two subject ontologies are included in the current version of DigLib-CI. The Computer Science ontology is based on the Computer Science Curriculum 2008 of ACM and IEEE/CS [5]. It defines the atomic knowledge units for the University courses and available research materials in the field of Computer Science and makes them sharable and reusable. Its current version includes approximately 300 concepts with their properties and relationships.

The Information Systems ontology has been under development using the Model Curriculum for Undergraduate Degree Programs in Information Systems of ACM, AIS and AITP [6].

The subject ontologies are designed as information sources describing the hierarchy and some other relationships between the main concepts in the discussed domains. A dictionary of synonyms has been developed with the purpose of providing the search engine with other viewpoints to the conceptual structure of the considered areas.

5.3 Main Features of the Search Engine

The purpose of the search engine is to provide adequate access to the complete palette of resources stored in DigLib-CI.

The search engine maintains several types of search and document retrieval within DigLib-CI. The user queries define restrictions on the values of certain metadata attributes of the required research or learning materials. Generally the search mechanism may be formulated as follows: the document descriptions included in all permissible for the user sections of the library are examined one by one and these descriptions which have a specific element (determined by the type of the user query) with a value matching the user query, are marked in order to form the search result. The matching process is successful if the value of the element or the value of one of its child elements is equal to the user query. The documents pointed by the marked descriptions are retrieved and the user is given an access to these documents and their catalogue descriptions.

The current implementation of the search engine supports four types of search and document retrieval:

- full search – search and retrieval of all available library resources, ordered by title, by author, by category, by date of creation or by date of inserting in the library;
- author search – search and retrieval of the documents created by a given author;
- ontological search – the search is performed in the semantic annotations of the documents;
- keyword search – the search is performed in the lists of keywords in the resource descriptions.

During the *ontological search* the user query is augmented with regard to the concepts searched out in the semantic annotations of the required library resources. The more specific concepts from each of the subject ontologies indicated by the user are added to the original one in the resulting query. Then the search engine retrieves all documents in the library containing in their semantic annotations at least one component of the augmented query. In this way the ontological search enables one to find documents described by ontology concepts which are semantically related to the concept defining the user query.

5.4 An Example of Ontological Search

Let us suppose for example that the user defines a request (a query) for ontological search concerning the concept “programming languages”. First an extension of this request will be generated. It will include all ontological concepts which are special cases of the concept given by the user (with respect to the ontologies indicated by the user). For this purpose, breadth-first search in the graphs that represent the ontologies will be performed, starting in each one from the concept chosen by the user.

Assume that the Computer Science ontology is chosen by the user. In this case the extended request (the augmented query) will include the concepts “programming languages”, “abstraction mechanisms”, “declarations and types”, “language translation systems”, “functional programming”, “logic programming”, “object oriented programming”, “procedural programming”, “programming language design”, “declaration models”, “garbage collection”, “type checking”, ... , “design goals”, “typing regimes”.

After that, a consecutive search in the catalogue descriptions follows. In this search all documents with descriptions that are juxtaposed with at least one element of the extended request, are extracted. In the current implementation each document appears as many times in the result list, as many elements of the augmented query are juxtaposed with its description (which means that the element <ontologyRefs> of the description includes a sub-element that has value, coincident with an element of the augmented query).

Figure 1 shows a screenshot displaying part of the ontological search results for the query “programming languages”.

The current activities are directed to the selection of a proper set of relationships between the ontology concepts that should be taken into account in the process of ontological search in addition to the hierarchical ones.

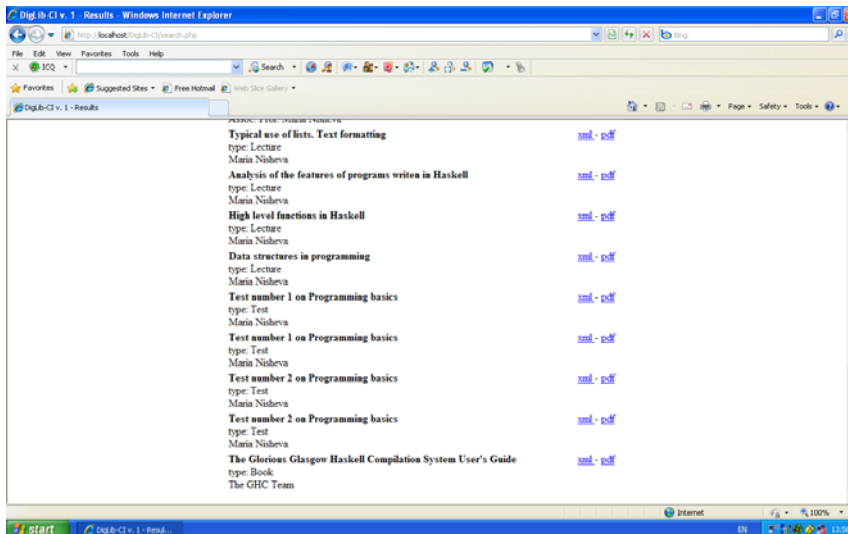


Fig. 1. Some search results for the query “programming languages”

6 Conclusion

The discussed architecture and working principles of the search engine of DigLib-CI are designed in order to support flexibility, interoperability and reusability. They could be applied in the implementation of the search engines of a whole class of academic digital libraries that provide semantics oriented access to their resources. In particular, a work in progress is going in order to enhance the considered search mechanisms and to utilize them in another project named “Information technologies for presentation of Bulgarian folk songs with music, notes and text in a digital library”.

Acknowledgements. This work has been funded by the Sofia University SRF within the “Methods and Tools Supporting the Lifecycle of Rich Digital Content” Project, Contract No. 168/19.04.2010. The study of metadata standards was accomplished within the implementation of Project VU-MI-204/2006 “Intelligent Sensor Systems for Security Enhancement” funded by the Bulgarian NSF.

References

1. Patel, M., Koch, T., Doerr, M., Tsinarakis, C.: DELOS: A Network of Excellence on Digital Libraries. D5.3.1: Semantic Interoperability in Digital Library Systems. UKOLN, University of Bath (2005)
2. Gruber, T.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies 43, pp. 907–928 (1995)

3. Guarino, N., Carrara, M., Giaretta, P.: Formalizing Ontological Commitment. Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), Seattle, Washington, July 31–August 4, 1994. Morgan-Kaufman, pp. 560–567 (1994)
4. The IEEE Standard for Learning Object Metadata. <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html>, last accessed on July 31, 2010 (2002)
5. ACM Education Council, IEEE Computer Society Educational Activities Board: Computer Science Curriculum 2008: An Interim Revision of CS 2001. <http://www.acm.org/education/curricula/ComputerScience2008.pdf>, last accessed on July 31, 2010 (2008)
6. Association for Computing Machinery, Association for Information Systems, Association of Information Technology Professionals: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems. <http://www.acm.org/education/is2002.pdf>, last accessed on July 31, 2010 (2002)

Catalogue Metadata in an Academic Digital Library

Pavel I. Pavlov¹

¹ Faculty of Mathematics and Informatics, Sofia University “St. Kliment Ohridski”
5 James Bouchier Blvd., Sofia 1164, Bulgaria
pavlovp@fmi.uni-sofia.bg

Abstract. The paper presents some aspects of an ongoing project aimed at the development of a methodology and proper software tools for building academic digital libraries. A particular functional model of a class of academic digital libraries has been proposed and analyzed. Some solutions of the variety of problems concerning the development of adequate mechanisms for semantics oriented search in multilingual academic digital libraries are discussed. An ontology-based approach is suggested in order to standardize the semantic annotation of the library resources and to facilitate the implementation of the functionality of the search engine. The emphasis of the discussion falls on the metadata used in the catalogue descriptions of the different types of research and learning materials in a particular academic digital library being under development at the Faculty of Mathematics and Informatics, Sofia University. These metadata are picked out in accordance with some internationally approved classification schemes, the traditional experience of Sofia University and the needs of the target groups of users.

Keywords: Electronic Publishing, Digital Library, Metadata, Semantic Annotation, Ontology, Semantic Web

1 Introduction

In the age of a networked society, IT in addition to their use in all spheres of human activity have been used extensively to record, store, and disseminate information in digital form. In particular, libraries are rapidly changing to meet the demand put on them. Their new generation should be developed as well equipped and interconnected *digital libraries*. The term *digital library* is diffuse enough to be applied to a wide range of collections and organizations, but, to be considered as a digital library, an online collection of information sources must be managed by and made accessible to a community of users. To this effect, many institutions are actively involved in building appropriate repositories of the institution's books, papers, theses, and other works which can be digitized or were “born digital”. In particular, universities and other academic institutions participate successfully in lots of projects directed to the development of suitable *academic digital libraries*. Academic digital libraries are committed to maintaining valuable collections of scholarly information. To this end, essential information resources should remain available and accessible into the future



– a real challenge in the cases of digital resources that are increasingly transient and at risk.

Besides its concrete institutional function, an academic digital library could play a significant role to:

- emphasize effective knowledge management and discovery as a key to innovation in learning and research;
- encourage the widest possible distribution of scientific, educational, and cultural information at the lowest possible cost;
- promote the needs and interests of diverse communities of users;
- inspire the creative application of technology in the service of scholarship and learning.

The paper presents a work in progress which is directed to the development of a methodology and proper software tools for building academic digital libraries. A special attention has been paid to the elaboration of tools for semantics oriented search in multilingual digital libraries. The study and the practical experiments are oriented to the development of DigLib-CI – a digital library with research and learning materials (articles, dissertations, monographs, lecture notes, textbooks, presentations, example program sources, data sets, quizzes, manuals etc.) created at the Department of Computer Informatics of the Faculty of Mathematics and Informatics (FMI), Sofia University, or especially selected among the scholarly materials freely available on the Web.

More precisely, the discussed project has the following main objectives:

- to explore the architectural principles, user interfaces and search engines of the most popular academic digital libraries;
- to study the various aspects of creation and integration of appropriate ontologies oriented to the contents of multilingual academic digital libraries;
- to define suitable metadata to accompany various types of scholarly and learning materials, taking into account the internationally approved classification schemes, the Bulgarian national traditions and FMI experience;
- to develop a proper framework for application of advanced information technologies and particularly Semantic Web technologies in building tools for semantics oriented search in multilingual digital libraries.

2 Architecture of DigLib-CI

DigLib-CI has the architecture of a typical academic digital library. It is designed in order to provide open access to various kinds of scholarly and instructional content, mainly in a wide range of subfields of Computer Science and Information Systems, developed or available at FMI. Its implementation is based on the use of specific Semantic web methods and technologies. The functional structure of DigLib-CI is shown in Figure 1.

The content repositories include various types of research and learning materials of different types (books, dissertations, periodicals and single articles, manuals, lecture notes, presentations, source code of computer programs, data sets, tests, quizzes etc.) in the areas of Computer Science and Information Systems, written in Bulgarian or in

English language. These library resources are available in various digital formats: pdf, html, plain text, doc, ppt, jpeg etc.

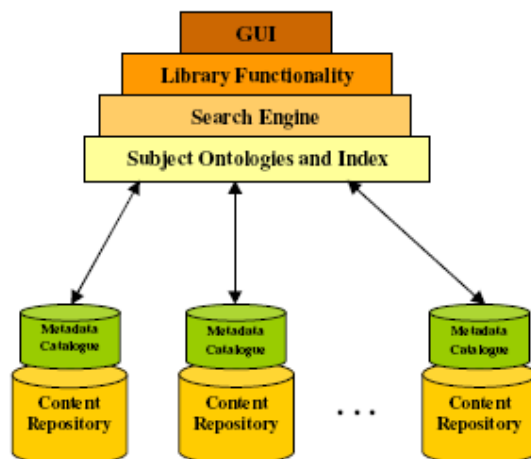


Fig. 1. Functional Model of DigLib-CI

The metadata catalogues are destined to facilitate the identification of the needed research or learning materials by the search engine. They contain descriptive metadata stored in XML format and support the reusability of all library resources and facilitate their interoperability.

The subject ontologies describe the main concepts of the areas of Computer Science and Information Systems, with their properties and different kinds of relationships among them. These ontologies play a significant role in the implementation of the full functionality of the search engine.

Two subject ontologies are included in DigLib-CI. The Computer Science ontology is based on the Computer Science Curriculum 2008 of ACM and IEEE/CS [1]. It defines the atomic knowledge units for the University courses and available research materials in the field of Computer Science and makes them sharable and reusable. This ontology includes about 300 concepts with their properties and relationships. The Information Systems ontology has been under development using the Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems of ACM, AIS and AITP [2]. The subject ontologies are designed in order to play the role of information sources describing the hierarchy and the other relationships between the main concepts in the discussed domains. They provide the search engine with a complete viewpoint to the conceptual structure of the areas of Computer Science and Information Systems.

The purpose of the search engine is to provide adequate access to the entire set of resources stored in DigLib-CI.

The library functionality and the user interface of DigLib-CI are designed in accordance with the expected needs and requirements of the basic types of users of

the library. The interface module provides adequate online access to the requested library resources and supporting software tools.

3 Types of Library Resources in DigLib-CI

The content repositories of DigLib-CI contain materials of the following types:

- books;
- periodicals;
- papers (articles);
- master's theses;
- dissertations;
- synopses of dissertations;
- lecture notes;
- presentations of lectures;
- presentations of seminars;
- descriptions of course projects;
- source code of educational software;
- data sets;
- tests and quizzes;
- unofficial materials.

In principle, the organization of the repositories of DigLib-CI is subject-oriented, therefore they contain research and learning materials of different types. Each repository has its own catalogue which includes proper metadata for the particular library resources.

4 Attributes of the Library Resource Descriptions

The library catalogues contain descriptions of the resources stored in the corresponding repositories. These descriptions consist of metadata (proper attributes with their values) which support the identification of the requested resources by the search engine. The metadata are stored in XML format and comply with the IEEE Standard for Learning Object Metadata [3].

Most kinds of research and learning materials have the following attributes: type of the material; author; title of the material; language(s) (human and/or programming one(s)); digital format; location; version; date of creation; completion status; restrictions on use; subject(s); semantic annotation – list of concepts from the proper subject ontology describing the Computer Science or Information Systems subfields and/or concepts covered or treated by the material. Learning materials have been characterized also by their educational level and the principal types of users for which the corresponding material was designed. Officially published research materials and textbooks are supplied with the corresponding bibliographic metadata.

All resource descriptions (i.e., all catalogue entries) consist of two equivalent parts in which the element values are texts in Bulgarian or English language respectively.

The search engine examines the corresponding parts of the descriptions according to the language of the user query.

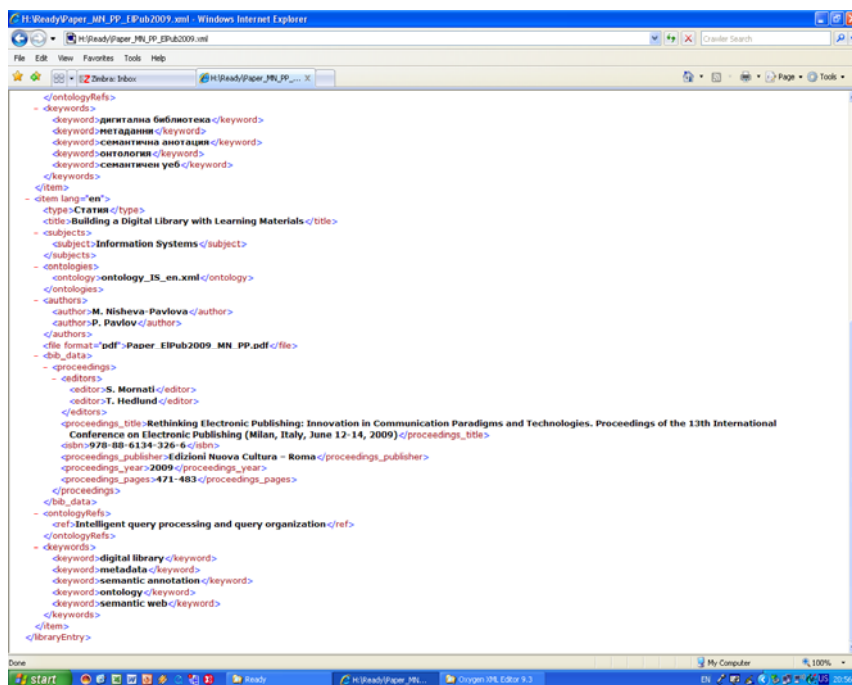


Fig. 2. Part of the catalogue description of an article

Figure 2 illustrates a selected part of the description of a paper (article) in XML format and Figure 3 shows a part of the DTD of the “paper” type library resource.

The element <bb_data> in the descriptions of officially published research or learning materials contains the respective bibliographic information (place of publication, e.g. periodical or conference proceedings; editor(s); publisher/publishing house; ISBN and/or ISSN of the book/periodical; year of publication etc.).

The elements <ontologyRefs> and <keywords> of the resource descriptions play the role of semantic annotations of the corresponding library materials. The values of the child elements of <ontologyRefs> are concepts of the appropriate subject ontologies (names of classes in these subject ontologies) which present most precisely the content of the respective document.

The concepts of the subject ontologies are too general from the point of view of the expectations of the typical users of DigLib-CI. Because of that, one can include in the resource descriptions additional lists of keywords which describe the content of the corresponding documents at an adequate level of abstraction. These keywords are set as values of the child elements of the <keywords> resource description elements.

```

<!ELEMENT libraryEntry (item,item)>
<!-- final version 14.08.2009 -->
<!ELEMENT item (type, title, subjects, ontologies, authors, file, bib_data,
ontologyRefs, keywords)>
<!ELEMENT type (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT subjects (subject+)>
<!ELEMENT ontologies (ontology+)>
<!ELEMENT authors (author+)>
<!ELEMENT file (#PCDATA)>
<!ELEMENT bib_data (proceedings|periodical)>
<!ELEMENT ontologyRefs (ref+)>
<!ELEMENT keywords (keyword+)>
<!--
<!ELEMENT subject (#PCDATA)>
<!ELEMENT ontology (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT proceedings (editors, proceedings_title, isbn, proceedings_publisher,
proceedings_year, proceedings_pages)>
<!ELEMENT periodical (periodical_title, issn, periodical_publisher,
periodical_year, volume, number, periodical_pages)>
<!ELEMENT ref (#PCDATA)>
<!ELEMENT keyword (#PCDATA)>
<!--
<!ELEMENT editors (editor+)>
<!ELEMENT proceedings_title (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT proceedings_publisher (#PCDATA)>
<!ELEMENT proceedings_year (#PCDATA)>
<!ELEMENT proceedings_pages (#PCDATA)>
<!--
<!ELEMENT periodical_title (#PCDATA)>
<!ELEMENT issn (#PCDATA)>
<!ELEMENT periodical_publisher (#PCDATA)>
<!--

```

Fig. 3. Part of the “paper” type library resource

The names of the relevant subject areas and the names of files containing the suitable subject ontologies have been assigned as values of the child elements of the catalogue description elements <subjects> and <ontologies> respectively.

5 User Interface

The library functionality and the user interface of DigLib-CI are designed in accordance with the expected requirements of the basic types of users of the library. The interface module provides adequate online access to the necessary library resources and supporting software tools.

The current version of the user interface allows one to formulate queries in Bulgarian or English language. It is intended for four types of users:

- FMI students – they may read/download textbooks, open lecture notes and presentations from all public sections of the library as well as all manner of other kinds of materials (monographs, dissertations, articles, periodicals, degree theses, lecture notes, presentations, exercises, programs, data sets, quizzes, tests etc.) from fixed public library sections;

- FMI lecturers and researchers – in addition to the students' access rights, they may upload materials to fixed public sections as well as create and update private sections and use materials in some of them;
- librarians (library administrators) – they have full access to all public resources of the library (may download and upload materials destined for all public sections of the library);
- interested citizen – they may read and download public materials of fixed types (e.g., dissertations, textbooks, open lecture notes and presentations).

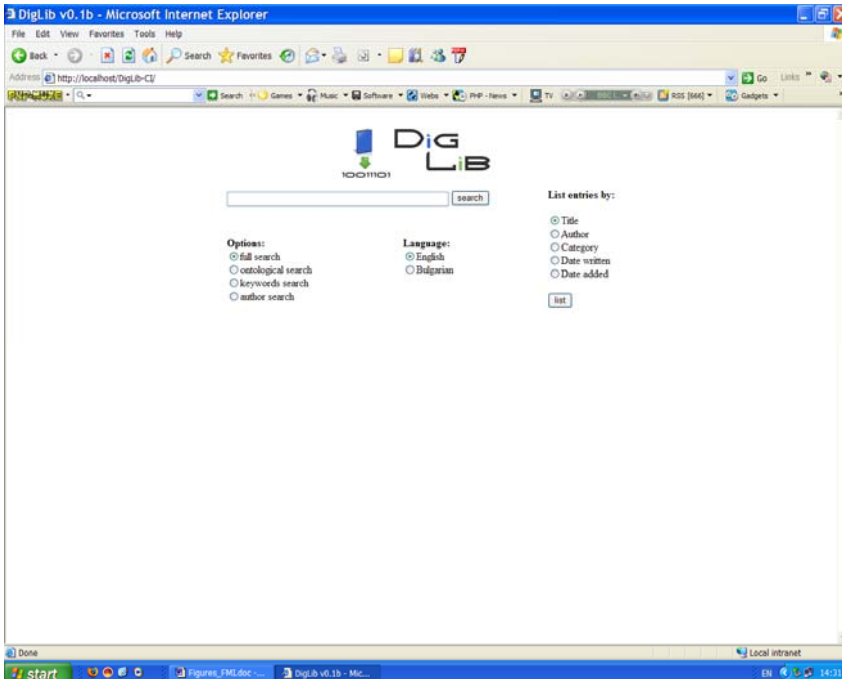


Fig. 4. User interface of DigLib-CI (search form)

All types of users of DigLib-CI may use the standard input interface which gives up convenient means for entering, editing and submitting queries for various kinds of document search and retrieval (see for example Figure 4). FMI lecturers and researchers as well as the library administrators may play the role of authors of library resources and catalogue descriptions, thus they have access to the author's part of the user interface. This part of the user interface provides for the authorized persons full access to the appropriate forms and enables them to enter and edit catalogue descriptions of all types of library resources (Figure 5). More precisely, the users may enter the values of some of the elements or pick out the values of others from previously drawn lists. In particular, the available subject ontologies can be properly visualized and the necessary concepts in them can be picked out as values of the child elements of the element <ontologyRefs>.

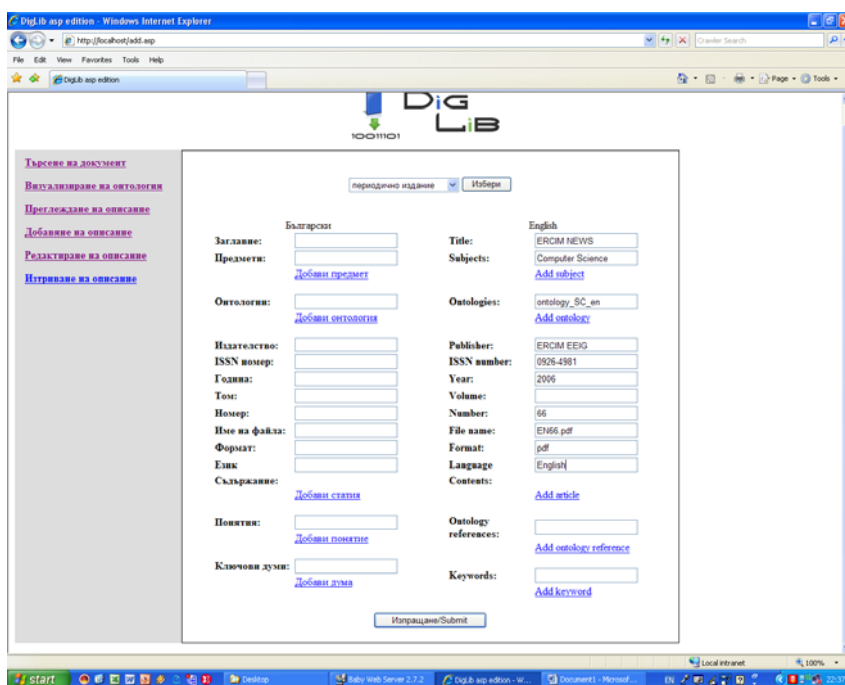


Fig. 5. User interface of DigLib-Cl (author's view – form for entering catalogue metadata of periodicals)

6 Conclusion

The most significant results of the discussed project obtained so far may be summarized as follows:

- A functional model of an academic digital library was proposed. This model provides tools for semantics oriented access to learning and research materials in various digital formats written in different languages;
- A prototype of DigLib-Cl – an academic digital library with research and learning materials in the areas of Computer Science and Information Systems, was developed.

As a main feature of the suggested approach to building academic digital libraries we could specify the provision of facilities for flexible semantics-oriented access to the library resources for users with various professional profiles and language skills. The selection of adequate catalogue metadata is one of the prerequisites for its successful implementation.

Acknowledgements. This work has been partly funded by the Sofia University SRF within the “Methods and Tools Supporting the Lifecycle of Rich Digital

Content” Project, Contract No. 168/2010. The author is thankful to his students Nadejda Apostolova, Anton Iliev and Klimentina Rousseva for their contribution to the implementation of the prototype of DigLib-CI.

References

1. ACM, IEEE Computer Society: Computer Science Curriculum 2008 (An Interim Revision of CS 2001), <http://www.acm.org/education/curricula/ComputerScience2008.pdf> (2008)
2. ACM, AIS, AITP: IS 2002 (Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems), <http://www.acm.org/education/is2002.pdf> (2002)
3. IEEE Standard for Learning Object Metadata, <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html> (2002)

Information Technologies for Presentation of Bulgarian Folk Songs with Music, Notes and Text in a Digital Library

Lozanka Peycheva¹, Nikolay Kirov², Maria Nisheva-Pavlova³

¹ Institute of Folklore, Bulgarian Academy of Sciences
lozanka.peycheva@gmail.com

² Department Informatics, New Bulgarian University
and Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
nkirov@nbu.bg

³ Faculty of Mathematics and Informatics, Sofia University
marian@fmi.uni-sofia.bg

Abstract. The paper presents the main aspects of an ongoing project aimed at the development of technologies for digitization of Bulgarian folk music and building a heterogeneous digital library with Bulgarian folk songs presented with their music, notes and text. The digitization of sound recordings and notations of folk music will facilitate the access to them, will provide their preservation in another quality and will present the opportunity for these recordings and notations to be used by a greater number of users through web-based access. The digital library will provide both digital preservation of the sound recordings and notations of Bulgarian folk songs and a possibility for new interpretations of the archaic Bulgarian folklore heritage. *Information technologies for presentation of Bulgarian folk songs* is an interdisciplinary project, which imposes a specific research approach, combining the plurality and dialogue of a multitude of viewpoints, positions and perspectives.

Keywords: Digitization, Digital Preservation, Digital Library, Bulgarian Folk Songs

1 Introduction

In the recent years informatics and information technologies have played an important role in a number of successful projects oriented towards the digital preservation of manuscripts, printed documents, photographs, audio materials etc. that are regarded as significant scientific and cultural heritage. At the same time the number of originally created in digital format documents, publications, books etc., which are important for the development of the public sphere and the industry have increased at high rates.

The availability of an enormous quantity of different materials on the web makes them difficult to access and thus it is necessary to organize them in the form of the so called digital libraries. In the past decade a great number of good results have been



achieved in the sphere of creation of digital libraries and the development of tools for access to their content. In particular, the project for an European digital library [1] was successfully implemented and in 2009 an initial version of a World digital library was launched [2].

At the same time the creators of digital libraries and tools for access to their content are still faced with numerous challenges. An example of such a challenge is the speed of fulfillment of user queries. Another challenge is the provision of maximum accurate and thorough responses to questions and queries of the users. As another serious problem can be viewed the necessity for definition and implementation of appropriate tools for search of a number of various by type and content materials accessible in different digital formats. In connection to that, the problems of the development of suitable methods and software tools for supporting the experts in different fields in the interpretations of the results from the search in big digital libraries is becoming an urgent matter.

The paper discusses the main activities planned within the scope of a project titled “Information Technologies for Presentation of Bulgarian Folk Songs with Music, Notes and Text in a Digital Library” which sets as its task to carry out a series of researches oriented towards the development of technology and an according supporting software tools for the creation and the adequate usage of heterogeneous institutional digital libraries. The practical tests and the pilot implementation will be performed on the basis of a collection of Bulgarian folk songs presented by notes, lyrics and music.

So far the most considerable contribution to the process of digitization, preservation and presentation of folk materials has been done by a project titled “Creation and presentation of a Bulgarian folk heritage collection”, that was developed by experts from the Institute of Folklore and the Institute of Mathematics and Informatics of BAS. In the last years new information technologies arise which improve and simplify digitization process (compare with [3]), and increase the potential for supporting more complex and flexible data bases and digital libraries.

2 Necessity of Digitization of Manuscripts (Music Notations and Lyrics) and Recordings of Bulgarian Folk Music

Bulgarian folk music is a valuable resource of cultural memory and is among the main characteristics of the national identity of Bulgarian people. Throughout the years the Bulgarian researchers of musical folklore have written down in lyrics and notes hundreds of thousands of musical folk samples (songs and instrumental melodies). Part of these music notations have been published, another part is preserved as manuscripts in specialized institutional or personal archives. The major part from the available music notations with Bulgarian folk music however is on paper. This, on the one hand, makes the access to them difficult and, on the other, bears the risk of their loss (especially, in the case where the notations haven't been published and exist in only one copy). Today's information technologies give us the possibility to digitize the existing collections of folk music records. This will facilitate the access to them,

will provide their preservation in another quality and will present the opportunity these records to be used by a greater number of users through web-based access.

The lyrics of the folk songs are also a valuable informational archive – they carry an affluent information for the lifestyle of the Bulgarian people during the centuries and for their mentality as well (religious beliefs, notions, cultural practices). The problem of the lexical composition of the Bulgarian folk song and its internal lexicological structure is still not solved. It is clear that the folk song lexis is completely different from the contemporary (and not only) colloquial and literary lexis, but this difference is still far from defined. The creation of the different types of indicators will be a starting point of many new scientific researches in different fields: linguistics, ethnology, ethnomusicology etc.

We propose a modern informational processing of recorded Bulgarian folk songs. The sound recordings of the songs in their authenticity were made mainly in the 1960-70s and are available only on old magnetic tapes. The authentic and unique field recordings collected in the course of decades from a number of Bulgarian folklorists (and other researchers) are preserved in the Musical Folk Archive in the Institute for Art Studies – the most significant collection of metadata for musical folklore in Bulgaria – mainly on magnetic tapes, the oldest of which date back to 1954. This type of preservation is not long-lasting and is in urgent need of digitization which will preserve the tapes for the future generations with the purpose of maintaining the cultural heritage of Bulgaria. The recorded tape is being damaged every time it is played – stretched, crumpled and demagnetized and often torn. This leads to deterioration of the material quality and to loss of particular fragments, or even whole samples and tapes. Thus unique copies are being lost forever. Contemporary information and communication technologies give the opportunity and the means to digitize these old music recordings, which is the only way for their preservation from a modern point of view and for making them accessible to scientific teams, educational institutions and the public. The process of digitization of old tape recordings is mainly focused on three technologies: transfer, mastering and conversion from Audio to MP3 files.

3 Some Achievements in Digitization of Folk Music

Currently on Bulgarian internet sites one can find music notes for Bulgarian folklore that are only scanned versions of the old manuscripts or parts of books. At the same time, there are a lot of musical archives all over the world, digitized with contemporary means and technologies. The emergence of an archive with Bulgarian folk music undoubtedly will provide for the incorporation of the Bulgarian cultural heritage to the world accomplishments in that sphere.

Special attention should be paid to the “Multimedia database for Bulgarian folk music” of the Institute of Art Studies of BAS, which consists of more than 15 000 records of song lyrics, analyses of ethno-musicologists, audio recordings, music notes and photographs and videos of Bulgarian folk music. Every record has a description – 23 fields of metadata and a few lines of the song lyrics. 1702 of the records also

consist of a brief musical recording of the original performance – “audio samples”, and 4633 – of a scanned copy of the piece of paper from the researcher.

In the summer of 2008 in Bucharest (Romania) a new European project was presented (with the participation of four countries – Switzerland, Germany, Hungary and Romania) on the development of information technologies for digitization of ethno-musicological archives (these are archives that preserve unique musical documents with great aesthetic and scientific value and serve the audio-visual memory and cultural variety). The project EthnoARC [4] provides a valuable experience, which reveals how it is possible to systematically present in linked archives the otherwise heterogeneous ethnomusicological archives in the cities of Berlin, Budapest, Geneva and Bucharest with the help of information technologies. This project is a practical confirmation of the increased necessity for the development of specific software tools for constructing of metadata in order to have an adequate, easy and convenient access to the unique musical folk documents preserved in the archives of the Institute of Art Studies and the Institute of Folklore in Bulgaria.

4 Digitization of Manuscripts and Books and Creation of a Database with Bulgarian Folk Songs

The digitization of notations is a specific and labour-consuming work. For the present moment the authors have around 500 digitized songs from the archive manuscripts of Prof. T. Dzhidzhev [5,6], which consists of field recordings of over 1000 folk songs from the Tracia region. Because the paper is getting old, it is important to preserve these songs in due time through their digitization in order to avoid the potential loss of information over time.

LilyPond [7] is a system for digitization of music records under GNU license. It is accompanied by a detailed documentation and an active mailing-list. The system is dynamically developing and its capacity is great – in practice it can be used for writing various notations – from orchestrations for big orchestras to Gregorian records of clerical music and special notations for bagpipe. The music records can easily be turned into PDF and PS files suitable for printing, EPS and JPG formats for inclusion in books and publications, as well as XML formats for communication with other software and databases. The syntax LilyPond is based on print preparation systems TeX and LaTeX. The programming language is C++, the software is structured in libraries, which are managed through Scheme scripts. There is quite a complex parser. The system is expandable for various specific needs with Scheme programming language, which is one of the two main dialects of the Lisp language.

Technology for coding of note records from original manuscripts with field recordings of Bulgarian folk songs

Despite the great LilyPond capacities, there are specific symbols and other elements necessary for notation of the Bulgarian national music. For example, the variations in some tones, bars and song parts, as well as signs for increasing or decreasing of the tone pitch cannot be written down with the available tools from the system. In line with the rules of the open code these elements can be proposed to the developers for inclusion in LilyPond and thus can become accessible for everyone.

Technology for scanning of printed books with BFS and OCR of the notation in them

This task encompasses the organization of technology for scanning of printed materials, mainly books, that contain Bulgarian national songs. After the scanning it is necessary to divide notes from lyrics and an identification (OCR) to be made on notes and lyrics separately. The difficulty in recognizing the notes is connected to the specific characteristics of BFS and obstacles will emerge for the lyrics also, because of the different dialects in the Bulgarian language. Besides, many of the words are written according to their phonetic sounding and not by the rules of the contemporary Bulgarian language.

Creation, support and completion of database for preservation of notes, lyrics and music recordings of BFS

The modern way of preservation of data is connected to their organization in structured (relational) database. The DB of Bulgarian national songs, which will be created by the project team, will consist of three basic components:

1. Notations of songs in LilyPond format;
2. Lyrics of songs together with additional information for the researcher, place and date of the research, name of the researcher, notes from the researcher;
3. Musical (MP3) files with an authentic performance of the song and musical files generated from notations with LilyPond system.

5 Information Technologies for Building and Using Digital Libraries

The aim of the research activities in this direction is to develop and adapt appropriate computer science methods, approaches, algorithms and tools for solving various problems concerning the analysis of folklore materials – methods for access to databases with music records and rhymed lyrics, analysis and effective (in particular, semantics oriented) search of specific regularities in the music and words of the songs etc. We expect to obtain some useful results in two main aspects: on the one hand, giving proper information technologies and software tools in order to assist experts of folklore in their professional activities; and on the other hand, to bring about new research topics in the field of Computer Science and Information Technologies, originating from the particular application.

Methods and tools for building heterogeneous digital libraries with folklore music

The objective of this task is to suggest a functional model of digital library with folklore music. Methods for building the particular components of such library and software tools for their integration will be developed as well. The experience and some former results of team members from the Faculty of Mathematics and Informatics of Sofia University [8] will be used for the purpose.

Semantics oriented search in digital libraries with folk music

This task has the aim to satisfy the necessity for the development and testing of a proper methodology for application of advanced information technologies (e.g., Semantic Web technologies) in building specific software tools for semantics oriented

search in the provided digital library with folklore music. These tools will support an adequate and flexible access to the complete palette of resources stored in the library.

Knowledge extraction from databases with Bulgarian folk songs

An attempt at formal conceptualization of a properly selected set of terms in the area of musical folklore will be made and a specific method for data mining and knowledge extraction from the database with Bulgarian folk songs will be proposed.

6 Conclusions and Future Work

The proposed database (the repository of the digital library) consists of the songs' notations in LilyPond format [7], the full lyrics of the songs, musical files, created by LilyPond and musical recordings of authentic performances (as far as such exist in the archives). This way the metadata will be extractable from the notes and from the lyrics of the song, for example bar measure, musical form, tone range, tone scale. Besides this, with the help of software tools scientific analyses will be able to be performed based on the songs notations and lyrics in the sphere of folkloristics, ethnomusicology and linguistics.

Our work is multidisciplinary, with the information technologies having the leading role for obtaining scientific results in connection with the solution of different problems arising during the process of creation of heterogeneous institutional digital library and the development of tools for the realization of different types of search in such a library, including semantics oriented search. Special attention will be paid to research dedicated to the creation of new methods and means for generating hypothesis and extraction of knowledge in the sphere of folkloristics. The scientific results in the sphere of ethnomusicology and linguistics would not be possible without sufficient collected information and software tools for their processing. For example, one ambitious task in folkloristics is to research the relationship between the lyrics of the song and the musical phrase.

The complete and up-to-date digitization – notes, lyrics, music – will complete and further develop what the experts from BAS have already achieved in terms of preserving the Bulgarian folklore. The application of European standards in the process will provide for the incorporation of the Bulgarian cultural heritage to the world's achievements in that area.

Acknowledgments. The project *Information technologies for presentation of Bulgarian folk songs with music, notes and text in a digital library* has been supported by Grant of the Bulgarian National Science Foundation under number DTK 02/54.

References

1. European Digital Library Project, <http://www.theeuropeanlibrary.org/portal/organisation/cooperation/archive/edlproject/>
2. World Digital Library, <http://www.wdl.org/>

3. Kirov, N., Problems and Achievements in the Digitalization of Bulgarian Folklore Music, (Proc. Int. Congress MASSEE'2003, September 15-21, 2003, Borovets, Bulgaria), Review of the National Center for Digitization 4, Faculty of Mathematics, Belgrade (2004), pp. 78–83.
4. Linked European Archives for Ethnomusicological Research, <http://www.ethnoarc.org/>
5. Peycheva, L. Stroitel na suvremennata bulgarska muzikalna folkloristika. Todor Djidjev na 80 godini [Builder of the contemporary Bulgarian Musical Folkloristics. Todor Djidjev's 80-th Anniversary]. Bulgarski folklore [Bulgarian Folklore], 2007, ISSN 0323-9861, 33/2, pp. 123–136.
6. Vlaeva, I. Todor Djidjev – terenistut [Todor Djidjev, the Fieldworker]. Bulgarsko muzikoznanie [Bulgarian musicology], 2009, ISSN 0204-823X, 33/3-4, pp. 90–101.
7. LilyPond ... music notation for everyone, <http://lilypond.org/>
8. Nisheva-Pavlova, M., P. Pavlov. Building a Digital Library with Learning Materials. In: S. Mornati, T. Hedlund (Eds.), "Rethinking Electronic Publishing: Innovation in Communication Paradigms and Technologies. Proceedings of the 13th International Conference on Electronic Publishing (Milan, Italy, June 12-14, 2009)", Edizioni Nuova Cultura – Roma, 2009, ISBN 978-88-6134-326-6, pp. 471–483.

Modeling the process of decision making using Petri nets and multi-agent systems

Elena Vlahu-Gjorgievska¹, Oliver Iliev², Nikola Rendeovski¹, Natasa Blazevska-Tabakovska¹

¹ Faculty of administration and management information systems
University "St. Kliment Ohridski" - Bitola, Republic of Macedonia,

²FON University - Skopje, Republic of Macedonia

Abstract. Interdisciplinary field of research called Socionics is the result of a common interest to sociologists and researchers in the field of Distributed Artificial Intelligence - DAI. Socionics research and monitor the formation and dynamics of artificial social systems with aims to develop intelligent computer technologies by using the paradigms of our social world, and vice versa, to use computer technology in the development of sociological models of societies and organizations. The example in this article concerns the decision-making process in public organizations. As appropriate techniques for modeling the characteristics of the system are used Petri nets with their advantages syntax, the graphical representation and formal semantics. Reference networks are used as the basic modeling technique for creating a base for multi-agent system. The relationship between micro and macro level is described from the perspective of computer science and from the sociological perspective.

Keywords: Multi-Agent Systems, Petri Nets, Socionics

1 Introduction

Socionics investigate and monitor the foundation and dynamics of artificial social systems. The research mainly focused on the genesis and dynamics of artificial social systems and human-machine hybrid societies.

Petri nets are based on fundamental duality of passive and active elements. They can be represented graphically, which is an advantage to use, especially interdisciplinary. Petri nets express important structural processes and concepts (processes of Petri nets are Petri nets themselves). Unlike other automata formalism, Petri nets allow modeling competition, the representation of simplicity, alternativity, parallelism, resources, actions and more.



Abstraction and intelligence are the characteristics that distinguish intelligent agents from other software tools, thus enabling software development to be conceptualized as a complex software system for collection of cooperative autonomous components that solve parts of a problem.

The basic challenge of socionics is to design high-modular multi-agent systems that can manage the societies of autonomous intelligent agents in open information society.

2 Socionics

Interdisciplinary area of research called Socionics is the result of common interest to sociologists and researchers in the field of Distributed Artificial Intelligence (DAI). Socionics investigate and monitor the foundation and dynamics of artificial social systems. Combining sociology and computer science is created a new research discipline which aims to develop intelligent computer technologies paradigms of our social world, and vice versa, to use computer technology in development of sociological models of societies and organizations. Researches are mainly focused on the genesis and dynamics of artificial social systems and human-machine hybrid societies.

In sociology there are two main views: micro and macro perspective. Micro perspective is oriented to the participants and from the socionics usually related to multi-agent systems. Macro perspective is structurally oriented and often leads to categorization of norms, values and rules to describe the social phenomenon.

The definition of the macro issues in sociology are significantly different from the meaning of same research in distributed artificial intelligence. Meaning, all types of social phenomena are located on the macro level, as opposed to individual agents that are locate on the micro level. The relationship between actions and behavior of the agent and the external influences and social structures often refer to the micro-macro link in the research of distributed artificial intelligence. The main topic for researchers from both areas is the problem how individual actions and structural rules affect the interaction of a set of agents. Therefore understanding the link between micro and macro level would lead to significant progress in the design of agents for

dynamic and complex agent based social simulations, and better cognition of human society. Micro-macro problem is directly linked to problems of coordination of the agents. This concept significantly contributes in distinguishing the distributed artificial intelligence from artificial intelligence. Therefore, micro-macro problem and modeling the macro aspects in agents theories is a central theme in the research of distributed artificial intelligence.

3 Petri Nets

Petri net is directed bipartite graph containing two types of nodes: places (passive elements) and transitions (active elements). Places are resources that may be available or unavailable based on certain conditions to be met. In classical Petri nets places and transitions exist exactly once during the occur (fire) of the network. If we generalize and allow multiple instances of each network, so that different instances can be separated and processed, referencing is allowed - meaning to use the network instances as a marker levels in a network. This leads to the formalism of reference nets, an extension of Petri nets.

Network instances provide a new level of dynamics. The system displayed by means of instances of reference networks has the ability to change as a result of add activities, change or movement of grid references. Without network instances would be almost impossible to change the status of system components without reorganization of the entire system. Reference networks are based on the paradigm of object-oriented programming in computer science. An object is characterized by its identity, encapsulation of the state and methods that allow external access to the state of the object. The well-known concept of inheritance should be mapped from the object-oriented programming to Petri nets.

The paradigm of network-to-network formalize themselves that aspect markers in Petri nets can be networks. These markers represent an advanced network concept because it indicates recursion in the theory of Petri nets. This opens the possibility of modeling hierarchical or recursive structures in a more elegant way which is the basis for modeling of sociological theory through reflective structures. Network which as a marker has a network is called a system network. Nets

markers are object nets. Nested (embedded) hierarchies with more than two levels and reference structures including the cyclic and reflective ones, are also possible. The inscription of the transition in the system network (which is associated with crossing object network) is called down-link. While its analog connection (up-link) is the corresponding inscription of the transition in the object network. The two links form synchronized channel in the reference networks. Two points that are part of synchronized channel can fire only synchronized (at the same time). When synchronized channel occur it is possible to exchange information between both transitions (in both directions).

There are several reasons why to use the formalism of Petri nets in modeling processes of decision making. On one side there is a wide range of analysis of patterns of Petri nets. On the other hand, Petri nets models allows collection of large amounts of data from simulation models.

4 Intelligent agents

The intelligent agent is a computer system located in a particular environment in which the agent perform individual actions in order to achieve its goals (given tasks that are designed for). In terms of agents, autonomy is an opportunity of the system to react without human intervention (or intervention by other agents) and to take control over own actions and internal state. While using the terms "autonomy" and "intelligence" refers to the context of community and technology. "Intelligence" related to "autonomy" is precisely the characterization of a intelligent agent.

Most powerful tool in solving the complexities of software development is modularity and abstraction. If the problem is complex, large or unpredictable, then most appropriate would be to develop a number of modular components that are specialized in solving different parts of the problem. In terms of agents approach to the problem, it is considered that the problem should give a larger number of smaller and simpler components, which is easier to develop and maintain component agent solutions. This makes the agent a powerful tool in making modular systems. When a problem appears, the agents in the system must cooperate among themselves and be well coordinate to resolve the problem. Autonomous intelligent software

agents enable development to be conceptualized as a complex software system for collection of cooperative autonomous components that solve parts of a problem.

In this particular example Mulan architecture is used. The architecture of multi-agent system Mulan is based on network-to-network paradigm, which is used to explain the natural hierarchy in agent systems. Mulan is implemented as a reference network system where the agent is an entity for processing messages. So, it must be able to receive messages, process them and generate its outgoing messages. It should be stressed that neither the content of the messages or way of their use is limited, while syntax structure should be constant (fixed). The basic model of agent include the encapsulation of the agents, the independence of their internal structure, ie access provided only through well-defined communication protocols. Providing the static interface is the main interoperability between agents.

5 Modeling the process of making decisions

One of the main objectives of this paper is to model a social problem with the model of Petri nets. Relations between concepts are shown more clearly with specific logic and Petri nets. In this way they can identify hidden defects in certain problems which would provide additional positive effects.

In this case, Petri nets are used to obtain an operating model, while providing simulation will rove (evaluates) social theory. Mathematical nature of Petri net allows the exact evaluation of formal analysis, which includes both: static and dynamic aspects. The further analysis are divided into two options: analysis by analyzing the network structure and verification of all conditions and dynamic analysis of Petri nets that provides an overview of the absence of actions that can be made, limit (on a sheet) in achieving a certain status, etc.

The main purpose of this example is to do research and analysis of inefficiencies and failures in decision making when hiring new staff in faculties. Often these commitments are made under previous regulations prescribed by the faculties. But there is a difference between the ideal application of these regulations and their implementation, and reality.

Recruiting new staff is according to the procedure: There is a need for professorial staff of a research area on the faculty. Academic Scientific Council (ASC) composed of professors of the Faculty, representatives of assistants and representatives of students discuss the announcing of the open competition. Following a positive decision (a majority of votes) the open competition is announced and all interested candidates who fulfill the conditions can apply. When the deadline expired council formed reviewal commission of teachers already employed in that or any other faculty and teach courses in the scientific area for which the competition is announced. Reviewal commission makes a report based on applications (backgrounds, levels of education, published works, teaching activity) of the candidates that applied and give a proposal (positive feedback) for the most appropriate candidate or candidates for the place, and the remaining candidates receive negative feedback. Council vote on the report of the Reviewal commission for candidates who have positive feedback. Candidates who'll get majority of votes are chosen for admission.

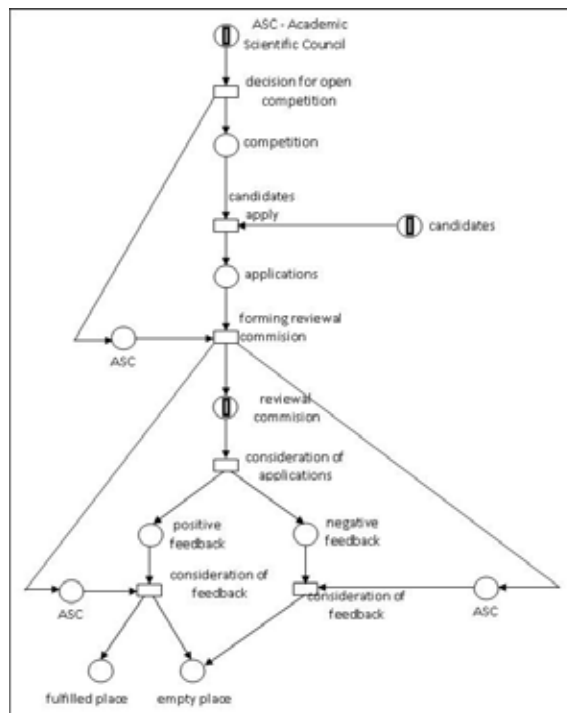


Fig.1.Macro perspective

In the interest of the faculty is to optimize this process and to achieve the desired results. But is important to mention a few things in the process: (1) candidates who have a positive feedback can be selected by council, (2) to make choices is necessary to get positive feedback and the majority of vote of the council.

Analyzing the empirical scenario actually council discussed the report of reviewal commission, accepting or not the application of candidates who have positive feedback. As a consequence of the real course of events occurs with the filing of a marker of the conditions *fulfill* or *empty place*. Empty place is not a desirable scenario by the faculty, taking into account that the engagement of new staff results with improvement of teaching (lectures) and research activities.

In sociology there are two main views (positions): micro and macro perspective. Micro perspective is oriented towards the participants and therefore often linked to multi-agent systems. Macro perspective is structurally oriented and often leads to categorization of norms, values and rules to describe the social phenomenon.

Figure 1 presents a macro model that shows the basic structures and relationships between agents.

While in Figure 2 is a representation of the micro model showing the structure of the processes and functionality of agents.

Both perspectives should be viewed as mutually relative one to another. Micro-macro link (micro-macro link MML) is especially important, because this link is seen as a key concept for construction of scalable multi-agenten system. The flexibility of this system is based on duality of micro-macro concept.

6 Conclusion

In this particular case, the example refers to the process of decision making in public organizations. As an appropriate technique for modeling the characteristics of the system, Petri nets are used with the advantages they offer as syntax, graph representation and formal semantics. Reference networks are used as the basic modeling technique and form the basis for multi-agent system. The same model development and agent protocols contribute to the flexibility of the

model. The link between micro and macro level is described from the perspective of science and also from sociological perspective.

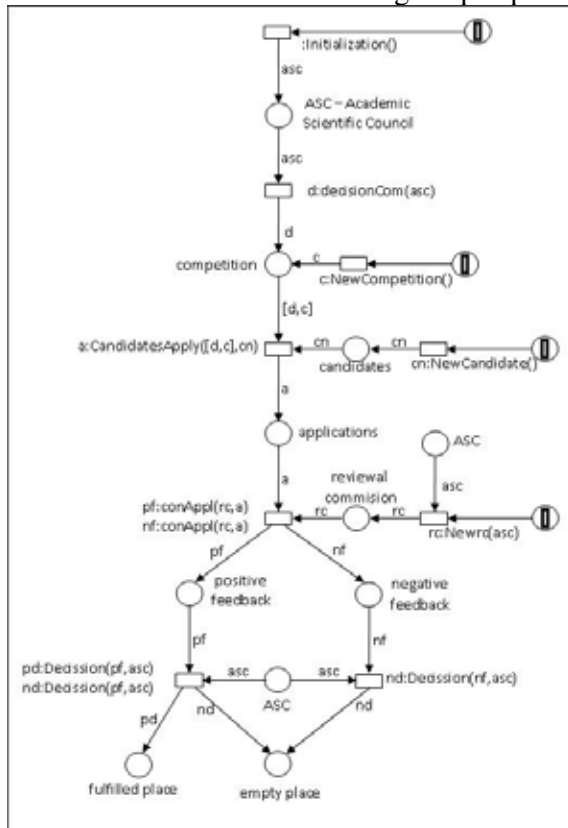


Fig.2. Micro perspective

Through the presented research, we are connecting computer and social research perspectives in computer-created societies, establishing models of multi-agent systems. These models can describe and analyze the processes in the computer societies as well as in human societies, while contributing their progress.

Presented models are only an initial step in developing a multi-agent system that would unite the achievements in socionics and both computer systems and social theories, respectively. The implementation of these models allows further analysis, implementation and use of their research results in other areas of computer science.

References

1. Köhler, M., Langer, R., Von Luumlde, R., Moldt, D., Rölke, H. and Valk, R. (2007): Socionic Multi-Agent Systems Based on Reflexive Petri Nets and Theories of Social Self-Organisation, *Journal of Artificial Societies and Social Simulation*, 10(1)3.
2. Fischer, K., Florian, M. (2005): Contribution of Socionics to the Scalability of Complex Social Systems: Introduction, *Socionics: Sociability of Complex Social Systems*, Fischer, K., Florian, M., and Malsch, T., (editors), volume 3413 of *Lecture Notes in Artificial Intelligence*, pages 1-14, Springer-Verlag.
3. Kummer, O.(2001): Introduction to Petri nets and reference nets, *Sozionik-aktuell*, 1.
4. Köhler, M., Moldt, D. and Rölke, H. (2001): Modeling the behaviour of Petri net agents, *International Conference on Application and Theory of Petri Nets*, Colom, J. M. and Koutny, M. (editors), volume 2075 of *Lecture Notes in Computer Science*, pages 224-241. Springer-Verlag.
5. Michael Köhler, Daniel Moldt, and Heiko Rölke. Modelling mobility and mobile agents using nets within nets. In *International Conference on Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 121-139. Springer-Verlag, 2003.
6. Cabac, L., Moldt, D. and Rölke, H. (2003): A proposal for structuring Petri net-based agent interaction protocols *International Conference on Application and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 102-120, Springer-Verlag, 2003.
7. Cabac, L., Dörge, T., Duvigneau, M., Moldt, D., Reese, C. and Wester-Ebbinghaus, M. (2008): Agent Models for Concurrent Software Systems, *Multi Agent System Technology*, Bergmann, R., Lindemann, G., Kirn, S., Pechoucek, M. (editors), pages 37-48, Springer-Verlag.
8. Köhler, M., Moldt, D., Rölke, H. and Valk, R. (2005): Linking micro and macro description of scalable social systems using reference nets, *Socionics: Sociability of Complex Social Systems*, Fischer, K., Florian, M. and Malsch, T. (editors), volume 3413 of *Lecture Notes in Artificial Intelligence*, pages 51-67, Springer-Verlag.
9. Dignum F., Greaves M. (Eds) (2000): *Issues in Agents Communication*, Springer.
10. Jennings, N., Wooldridge, M. (1998): *Agent technology: foundations, applications, and markets*, Springer-Verlag, New York.
11. Petrie, C. (1996): *Agent-Based Engineering, the Web, and Intelligence*, *IEEE Expert: Intelligent Systems and Their Applications*, Volume 11, Issue 6, pp 24-29.

APPLICATION OF THE INTRANET IN THE ORGANIZATIONS AT KNOWLEDGE MANAGEMENT

Blazeska-Tabakovska Natasha, Manevska Violeta, Vlahu-Gorgievska Elena
Faculty of administration and management information systems
University "St. Kliment Ohridski" - Bitola, Republic of Macedonia,

Abstract. In this paper will treat the problem of the use of information systems in daily operation of organizations in society as a tool for knowledge management. To successfully develop a system for managing knowledge in an organization requires the release of creativity and talent of its employees. Powerful tool for achieving this goal is the Intranet, which will enable interconnection, effective communication, creativity and recording of intellectual capital in the organization. Implementation of intranet as part of a system for managing knowledge in an organization is only part of the solution. Application of Intranet from a technical standpoint it is not difficult especially if you take into account existing information resources that are available, but difficulties occur in the process of creating habit for use by employees. Effort should be made in providing an environment that will be ready for locating and sharing knowledge by building a culture of sharing knowledge, in which cooperation is high valued and placing priority on intranet-based software tools for managing knowledge. This article presents the research of application of the Intranet in knowledge management in organizations in Macedonia.

Keywords: Intranet, Information system, knowledge management

1 Introduction

The management of knowledge is a relatively young science, in literature and also meets management of knowledge organizational intellectual capital. In recent years many organizations in this organizational capital has been paid great attention. Last year many major companies in America and Europe have opened a new position of head of department or head knowledge of organizational capital or responsible for the department of intellectual capital.

The question arises why this science just now. Several global trends have a significant impact on the development of this science, such as:

- Globalization of the economy, which made a strong pressure on companies to increase the adaptability, innovation and rapid change;



- Awareness that innovation is the product of group work and effective work that largely due to shared knowledge of how machines work, how to improve their work, which are demands and wishes of clients and how to improve service to them;
- Awareness of the importance of specialized knowledge embedded in organizational processes and routines;
- Awareness that knowledge is an obvious factor in the production and its role on the growth rate of the market;
- computer network as a cheap and accessible tool for learning from each other.

Working in an period of rapid technical and technological development at a time of rapid change and assorted customers, one of the main prerequisites to gaining the competitive advantage of companies is knowledge, or to be more precise *what companies know, as they use their knowledge and how fast they can acquire new information.*

Simply organizational knowledge is important for successful operation of the company; it's oxygen to company to provide competitive advantage. Once the organization will establish a policy of sharing knowledge management, managing will not have to control how they do not control the breathing of his employees. Knowledge will flow naturally, carrying the constant company a higher level in the market. But there are many threats to the extremely worth knowledge: to be stored somewhere in the organization to forget that it has; to be lost through organization; to leave in the minds of employees or the worst option to become obsolete and / or unused.

Important questions that should be given the organization: the flow of knowledge across the organization, how it is organized, growing and multiplying, how to acquire new knowledge, what happens when people leave the organization, whether their accumulated knowledge gone with them, how the expertise of one or two employees in the organization can grow in knowledge of entire organization.

2 Importance of shearing knowledge in organization

According to Michel Foucault "can not be practiced power without possessing knowledge. It is impossible knowledge not to cause power. "As long as knowledge and power are in interpersonal

relationship, knowledge management will result in social and economic power.

Knowledge is the only resource that can be sold, be used to give, and again the next day to dealing with it in any way and yet continued to possess. In the context of continuing this process the knowledge should be renewed, to add new value, to expand and create more knowledge. Today at a time of rapid technical and technological development the knowledge can quickly become outdated. It constantly changes and it must be renewable and to drop deprecated old knowledge. New knowledge and expertise in organizations is created by teams and groups of employees and it is sharing throughout the organization. If it is left to occur at the individual level, is almost certain that the individual accumulated knowledge quickly becomes obsolete and useless for a day, week or at most several months. If the knowledge not flows, not increased, over time becomes obsolete and useless. It is the same that the money is not invested, they lose their value until they become worthless. Towards this, the knowledge which flows, shares, builds and amended, generate new knowledge.

Although in the nature of the people is to share knowledge, when they are a part of an organization there is additional elements that hinder or even disable sharing, such as competition, fragmentation between loyalty to the organization, loyalty to the team and ultimately loyalty to their own career. The base of the culture of knowledge sharing is a trust. Important factors which can lead to undistributed knowledge is distrust or perception that others who shares that knowledge are enemies, competition, and can use this knowledge against us. Therefore the efforts should be targeted at providing an environment that will be prepared to locate and share knowledge by building a culture of sharing knowledge, in which cooperation is highly valued and placing priority on Intranet based software tools for knowledge management.

Knowledge management tools emerged somewhere in 90s, but they do not fully answered the needs for this purpose. Reasons were many. One of the reasons was the individual characteristics of the various entities and accordingly different approach to managing knowledge.

Starting from the fact that very important for the management of knowledge is increased communication between employees within the organization and increased communication with suppliers,

customers or service users and the overall external environment, the tendency is to provide tools that will improve communication. It can be found thinking and messages to those responsible for supplying the market with tools that will assist knowledge management to be as simple as possible for use. The reason is simple. These tools are intended for broad use for a wide range of employees not only for specialists in the field of information technology and IT managers. Simple tools will help employees to perform their tasks more easily, be bowed to their use and more willing to share information which will be gradually abandoned the use of advertising boards.

3 Application of Intranet to facilitate sharing of knowledge

For successful development of a system for knowledge management in an organization requires the release of creativity and talent of its employees. Powerful tool for achieving this goal is Intranet, which will enable interconnection, effective communication, creativity and capture the intellectual capital within the organization. Implementation of the intranet as part of a system of knowledge management in an organization is only part of the solution. Implementation of the intranet from a technical point of view it is not difficult especially if you take into account existing information resources that are available, but difficulties occur in the process of creating habits for use by employees. Another thing that should be practiced is improving the existing intranet. There are many possibilities: removing outdated content, adding applications for collaboration, implementation of new social media initiatives. Definitely is confirmed the view that intranet as a collaborative tool facilitates the process of the four factors of success: *communication* - sharing news and events; *business processes* - technical characteristics; *cooperation* - discussions of working groups between different departments; *management of knowledge* - to facilitate the process of sharing of knowledge among users, sharing of good practice or practices which could be extracted positive experience. Today intranet is far from what was in the past, a tool for communication. Today it has strategic importance. When it is started deploying intranet one tool may facilitate its use in the context of knowledge management, it is a system for managing content (content management system-CMS). That tool will allow organization of

documents and will help to find which content where is stored. Can often be heard making the recommendation, to write your own CMS because he would be more appropriate to the requirements of the organization that is better addressing the needs of users as well as better understanding of the team for its development. Moreover is recommended if the organization has the necessary expertise and has sufficient time to make it developed within the organization.

All these recommendations addressed to organizations contain a single goal, this system to meet the specific needs of the company. The first is *content management system* - function the most vital. It refers to storage of various content such as documents and new records in the database and displays them whenever the user requested. Activities such as adding, removing or modification content should be simple for users. A second characteristic that should be provided through individual approach to building an appropriate CMS is *authentic users* - defined access level or degree of restriction of access to certain areas. Some areas of intranet pages can be available to add, processing and modification content while others may be available only for employees in that department. *Identical intranet pages* is the third important feature of this system and it aims the displayed image data requested on screen to be identical for all data regardless of which application has been set. There is another very important matter that must be taken. The intranet can be implemented well, to incorporate within itself all the recommendations, to build experience from the best practices, to be released from all of the bugs, tested in several focus groups to gain feedback that meets the needs of users, but despite all attempts to "lured" consumers to use it can absence its application. The reason may be lack of system knowledge and training or simply work on a negative perception.

There are several things that may help to propitiate the employees. *Permission to create content* - Some intranet is unidirectional: They broadcast information about users, and that's all. But if we consider that whenever something feels as ours, when it has a personal stake appears greater interest, imposes the need for involvement of the users of the Intranet to become active participants in creating content. To ensure credibility of the set content is only needed the approval process of the content.

Setting of interesting content and / or video clips that are not directly related to work, can interested employees to explore other content from the site.

Another possibility is the use of *micro blogs*. In this way, users will be informed by intranet for new content or features versus their bombardment of information with sites that can cause intimidation while avoiding the use of the intranet.

At some point after implementation of intranet it is necessary the *focus of employees to put on learning*, to give future users useful information, tips, how to do some activity and the system and documentation.

Using the tools of social networks has a major impact in strengthening the strong corporate culture. You need to have different access to different users of the intranet. Thus users will be able to obtain more information directly related to their daily work.

Speaking of the classifieds should not be too much energy spent on talking about its technical characteristics but also to focus on consumers and its impact on processes.

4 Application of Intranet in knowledge management in organizations in Republic of Macedonia

The survey, whose problem was "the application of the Intranet in knowledge management in organizations in Republic of Macedonia" was, included several different organizations in more cities and more activities: public enterprises, industrial companies, banks, insurance companies, telecommunications companies, media houses, brokerage houses, shops, hotels and others. Data was collected through questionnaire and interviews with representatives of competent organizations.

The objective of this research is to answer the following questions:

1. How is the application of information systems in organizations in Macedonian organizations?
2. How much is paid attention to the management of knowledge?
3. Do organizations have intranets?
4. What is the impact of intranet on the management of knowledge?

These goals were guides in this research and during its application it is taken care to establish a complete picture of information system in individual organizations. Data analysis showed that very few Macedonian organizations apply the system of management knowledge. 99% of organizations surveyed have implemented information system which is used for different purposes, but only 2% said they have a system for managing knowledge. As possible reasons for this situation can allocate more:

- Top management is still not sufficient commitment to this issue, it is not their priority resulting in weak support;
- strict hierarchical and inflexible structure in Macedonian companies;
- fear of sharing knowledge, as fear of competition and fear because of lack of readiness to use the system for managing knowledge;
- lack of awareness about the benefits of applying this system;
- reliability of data.

Only 5% of the organizations included in this study reported that they have implemented intranets, which was not enough to require a connection between the intranet and the management of knowledge. All this raises the conclusion that greatly organizations in Macedonia to improve their business processes use information systems, whose maintenance and recovery care employees in organizations. Many of the organizations surveyed have internet connection, but only a small part of them have a private-network intranet, which leads to the conclusion that much more oriented towards communication with the outside, as opposed to communication in the organization and thus share knowledge. Considering that knowledge management is a relatively young science is not a surprising fact about this situation in the organizations surveyed. A positive element is the willingness of managers in the future in its focus to set management system of knowledge.

References

1. Alle V.: *The knowledge Evolution- Expanding organizational intelligence*: Butterworth Heinemann, USA (1997)
2. Liebowitz, J.: *Building organizational intelligence*: Wisdom Builder, USA(1999)

3. Balaban, N., Ristic, Z., Durkovic, J., Trninic, J.: *Informacioni sistemi u menadzmentu*: Branko Dzonovic, Beograd, (2005)
4. Laudon, K., Laudon, Dz.: *Menadzment informaciski sistemi*: Ars Lamina, (2010)
5. Intranet and knowledge sharing, www.steptwo.com.au

Young Scientists Section

Development of Accounting Services for Grid

Radoslava Goranova

Faculty of Mathematics and Informatics, University of Sofia “St. Kliment Ohridski”,
James Baucher 5, Sofia, Bulgaria

radoslava@fmi.uni-sofia.bg

Abstract. In this article we present an approach for development of web services for accounting by using IBM Data Studio Developer. G-Lite is a grid middleware that provides access to data storage and computational resources. The accounting services of g-Lite middleware are based on R-GMA relational database, where all accounting information is stored. The end user can access limited R-GMA information by his certificate. This causes inconvenience for development of grid application, in cases when more verbose accounting information is needed. The object of this research is to describe development of web services for g-Lite middleware accounting and their usage in more complex sequences of tasks.

Keywords: Grid Computing, Webservice, g-Lite Accounting

1. Introduction

The Grid [1] is a hardware and software infrastructure for coordinated use of distributed resources, shared by different institutes, computational centers and organizations. The Grid resources can be sensors, computational resources, storage resources and others. The resource sharing is coordinated. By sharing we mean not only file sharing, but direct access to computers, software and data. The access to the Grid resources is agreed with policies. They define who can use the resource, who share the resource and also the conditions under which these resources are shared. This part of the infrastructure is covered by Grid security.

Grid middleware is software environment, which coordinates and integrates Grid resources, services and virtual organizations. g-Lite is a Grid middleware that provides services for effective data storage and data transfer, for security, monitoring, accounting and computation. The middleware is developed in order to meet, mainly the needs of the high-energy physics for computational and storage resources.

g-Lite security supports authentication through X.509 digital certificates, credential delegation and single sign-on.

The accounting and monitoring are part of the information system of the environment. These services give information about sites availability, statistic information for number of running jobs, information for Grid services availability. The use of these services, improve the middleware reliability.



Access to the services is possible through Grid certificate – X509 digital certificate, associated with user identity. Without certificate the information services of the middleware, including and other main services as data storage and computation are not accessible.

The aim of this research is to provide an approach for development of web services for Grid accounting, based on IBM tools. By using tools features we will demonstrate the approach and show how accounting services can be exposed as web services.

2. g-Lite Middleware

g-Lite [2] is a Grid middleware, which is designed and implemented for EGEE Grid infrastructure and is tightly specified for the need of the project. The middleware provides Grid services for resource brokering, computing, information and storage. On Figure 1 is shown g-Lite services grouped logically into five groups:

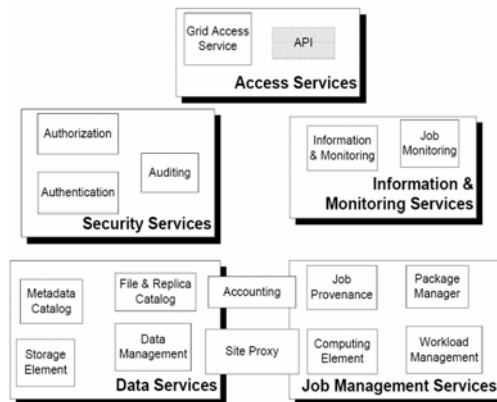


Figure 1. g-Lite Services

Security services, provides functionalities for

- User or service identification;
- Grant or revoke of user access to the service or the resource;
- Provides information for analysis, when security violation events occurred.

The middleware uses two approaches for implementing Grid security. First one is based on Grid Security Infrastructure (GSI) with Public Key Infrastructure (PKI) and the second one on Virtual Organization Management Service (VOMS) technologies.

PKI provides mechanism for certificate creation, management and verification. Certificate is a file, which combines digital signature with a Certification Authority public key and user identity.

VOMS is a grid technology which provides mechanism for user access to the resources. It provides interfaces for users to apply for VO membership, for administrators to manage the users, and for other Grid services to query stored information (user lists, roles etc).

The monitoring and information services provide mechanism for task monitoring, resource discovery and services' information retrieving. They are realized by two components:

- The Berkeley Database Information Index (BDII) – is a standard LDAP database, which is updated by external processes. The update includes extracting and merging information from different sources. g-Lite provides standard clients for access to the LDAP database.
- The Relational Grid Monitoring Architecture (R-GMA) – is information and monitoring service. The service is based on the model Producer-Consumer-Registry. The consumer requests for data, the producer provides the data. This service supports centralized registry, which mediates information between consumer and producer. The producer request registry to publish data. The consumer request registry to find the producer. The data itself are passed from producer to consumer directly.

The job management services include:

- Computing Element (CE) – computational resource, usually clusters of computers, but can be supercomputer or single computer. Provides information for working nodes and general interface for job submission and execution.
- Workflow Management System (WMS) – schedule the tasks on available computing elements, according user requirements. The user can request memory, processors and parallel execution of the job. WMS also tracks the job execution, by tracking and logging system
- Accounting – Provides information for the number of the jobs for a site, for users and their jobs and other statistical information.

The data services include:

- Storage Element (SE) provides access to storage resources, single disk or hard disks missives.
- File catalog and replication – save information, about data location.
- File Transfer Service (FTS), allows data to be copied from given storage to another. All services work with one or group of files.

Cluster of computers, with installed basic g-Lite services as CE, SE, BDII is called Grid site.

3. R-GMA and Accounting Services

In the previous section we gave a brief description of the services that g-lite middleware provides. The aim of this section is to give more detailed information about two of the services: R-GMA and Accounting and motivation why the development of web service for them is needed.

R-GMA [3] is an implementation of the Grid Monitoring Architecture that models the information infrastructure of a Grid as a set of Consumers, Producers and a Registry. R-GMA imposes a standard query language on this model – so producers publish tuples with an SQL insert statement and consumers query them using SQL

select statements. R-GMA also ensures that all tuples carry a time-stamp, so that monitoring systems are inherently supported.

R-GMA services permit or deny access to resources to a user on the basis of credentials held by that user. Credentials are extracted from the user certificate used for authentication. As we already mentioned, the user certificate is X509 digitally signed file, with user identity. For job submission, the Grid users use temporary file containing user's credentials, user's membership of virtual organizations and other information. This file is called proxy. Each credential has a name such as the Distinguished Name. The credentials found in the proxy certificates gives also information about user's membership of groups within a virtual organization and his roles.

R-GMA conforms to the Web Services Architecture. The services it provides has a well defined set of operations described into WSDL document. The R-GMA Web Service is currently implemented using a SOAP/HTTPS binding and uses Apache Axis software running in the Apache Tomcat container (Figure 2).

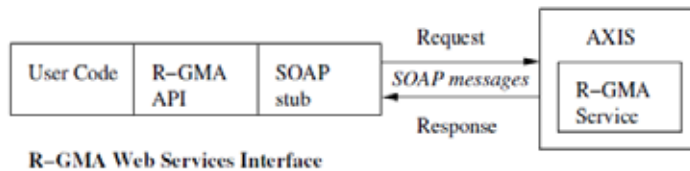


Figure 2. R-GMA Web Service Interface

The R-GMA browser can be used to access R-GMA service. It provides the following functionalities:

- View definitions of available tables in the schema;
- View producers that are publishing to a table;
- Submit a mediated query on a table;
- Submit a query to specific producers of a table;
- Publish data to a table.

The query language supported by R-GMA is a subset of SQL. SQL select statements are restricted by three components: the R-GMA SQL Parser, limitations on continuous queries, and limitations imposed by external components.

Continuous queries must be of a form which can be evaluated on each tuple in isolation. R-GMA parser does not accept nested select statement after keyword *SOME*, the keyword *HAVING* after a table name or *WHERE* clause, use of more than one column functions.

We have to mentioned, that limitation mentioned above are new. Recently, the limitations of using more complex SQL select statement were even more restricted. Also currently APIs for R-GMA are in a state of change to a new one. So the R-GMA service is in state of modifications and improvements.

The accounting services of g-Lite middleware are based on R-GMA relational database. The service is realized by Apel software.

The Apel [4] is composed of two components: The Log Parser and Publisher (Figure 3). The Log Parser interprets log files to extract job information and publishes it using R-GMA. Specifically, it processes the LCG gatekeeper logs, the system message logs and PBS event logs. Extracted data is then stored within a MySQL database. The Apel Log Parser also makes LDAP queries of the Computing Element to obtain the CPU performance figures for the worker node clusters and sub-clusters.

The Publisher is used to generate accounting records derived from the parsed logging data. The accounting records are then published into R-GMA where they are then collected by a central accounting server which, aggregates records from all sites.

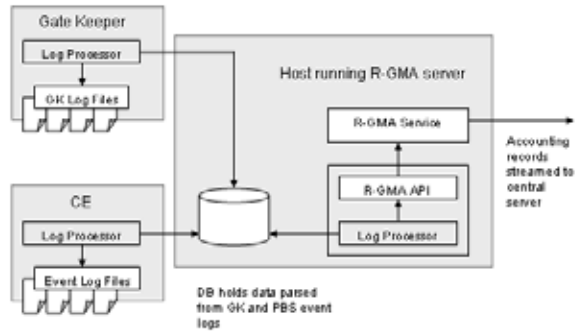


Figure 3. The Accounting Service

As we saw there is API that the g-Lite middleware provides for development of clients for access to R-GMA service. There is also R-GMA browser, which is convenient way for querying the R-GMA database. The problems in these cases are two. First the user has to be familiar with the SQL language in order to write simple query and second, related with the first, the user has to know all SQL limitation of R-GMA service. The inconvenience here is that if the user needs more verbose statistical information, for example grouping, counting, summing and etc., he can difficulty get it.

The other aspect is that all services that g-Lite provides, require Grid user certificate in order to access them. This causes inconvenience for development of Grid application, in cases when some statistical information is needed. The aim of our research is to apply an approach, where the access to accounting data is exposed by web services, without implementing g-Lite security. The approach is based on some IBM Tools for web service development, deployment and registration into common registry. The security matters are not in the scope of our research.

4. Approach Walkthrough

IBM Data Studio [5] is Eclipse based integrated development environment. It provides functionalities for creating and testing database objects, queries, database logic, and pureQuery applications. IBM Data Studio has integrated query editor for

SQL that can be used for database development, for application and queries optimization and for stored procedures testing.

Data Web Services (DWS) is the set of functions included in the IBM Data Studio that provides the capability to generate Web services-based access to DB2 or Informix database. DWS facilitate the development, deployment, and management of Web services. It is mainly a bottom-up approach to expose existing functionality (stored procedures or SQL statements) as Web service operations. By using data web services, you can take statements, such as SELECT, and generate Web services without writing a code. Data web services also provide a full Web-service interface, which includes support for SOAP and REST-styled bindings. The generated Web services are packaged in the form of a ready-to-deploy Web application, which can be deployed to supported application servers. The DWS supports SOAP over HTTP and WSDL generation that contains a description of the Web services.

The approach we proposed is based on development of web services by using IBM Data Studio developer. In our scenario we have all needed data into MySQL database. As we mentioned R- GMA conforms to the Web Services Architecture. So services that provides are service-oriented and can be easy and conveniently used. But for access to them the user has to possess Grid certificate. In order to avoid this, we are developing web services, which provide direct read-only access to the database, where data are stored. The case is examined for monitoring and accounting of a single Grid site. Because R-GMA and APEL use MySQL for data storing and IBM Data Studio is developed to work with DB2 or Informix we had to migrate the date from the one database management system to the other. We used a migration toolkit, in order to do that. IBM Migration Toolkit [6] provides a full-functioning GUI interface and supports migrations from source databases: Oracle, Microsoft SQL Server, to target databases: DB2 for Linux, UNIX, and Windows.

Once we have migrated data, we create web service by writing simple SQL queries. IBM Data Studio interface is shown on Figure 4.

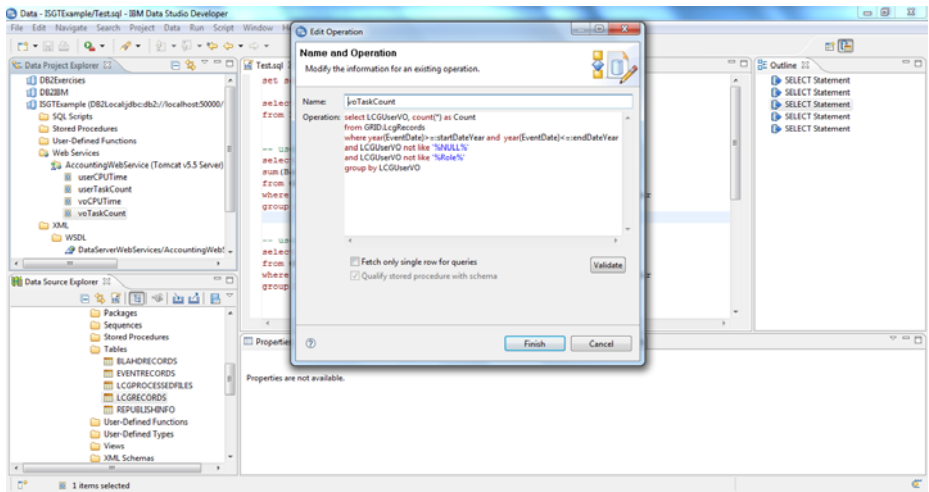


Figure 4. Web services development

When service is developed we build deployable files and deploy them on Tomcat 5 application server. The application server has to support JDK1.5 and also Axis and SOAP libraries. You need to copy i.e. soap.jar and axis.jar into application server's library directory. The application deployment to Tomcat server is easy (Figure 5). You have to copy created archive file into the server's web application directory.

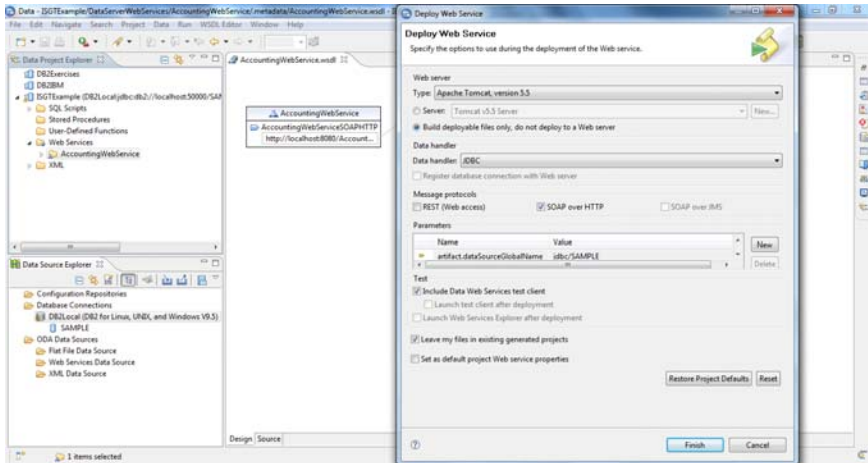


Figure 5. Web service deployment

Once the services are deployed you can test them. IBM Data Studio Developer generates simple clients to test the developed web services. This feature is very convenient, especially when quality of services is required. To make deployed services discoverable you have to register them into registry. For this purpose we use IBM Websphere Service Registry and Repository (WSSR).

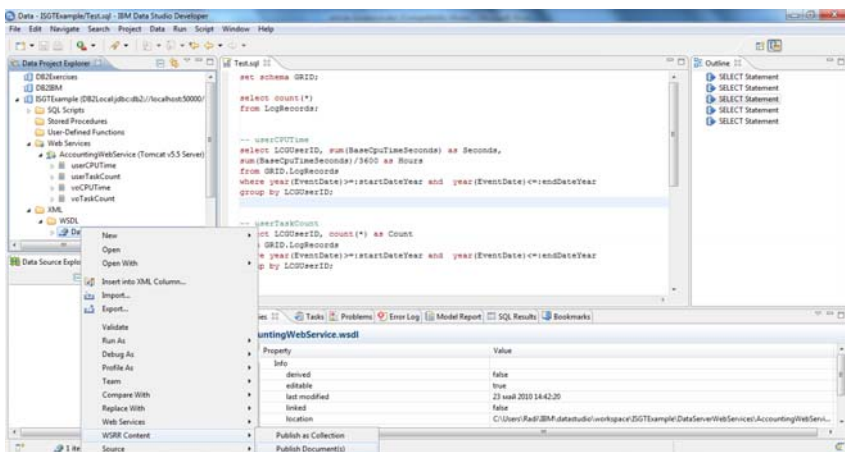


Figure 6. Web service publishing

WSRR [7] is service registry and repository that provides Eclipse based plugin for service publishing. IBM Data Studio integrates very well with provided WSRR plugin. That gives you opportunity easily to register developed service in the moment you finish with its development. On Figure 6 is shown, how easy is service registration, just with a click of a mouse.

After services are registered they can be browsed by web browser that WSRR provides. It is shown on Figure 7.

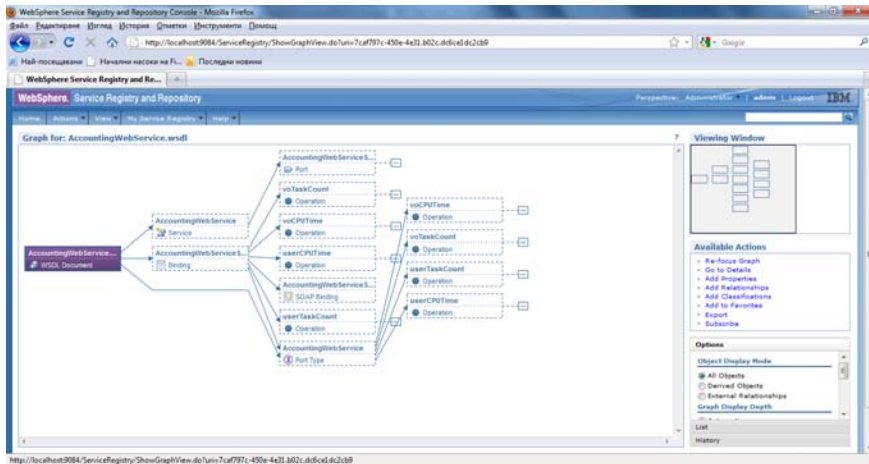


Figure 7. Graph of registered service into WSRR Browser

5. Conclusion and Future work

We show an approach for web service development, based on IBM Data Studio Developer tool. The developed services are registered and discoverable. That makes them easy to be used in more complex applications or more complex sequences of tasks – the processes. The development of the services on service-oriented manner, guarantee that they are loosely-coupled, easily reused and composable. The services' capabilities to participate into task composition or process is a great feature especially for those scientific arrears where business processes are needed.

The approach we proposed can be extended with services developed by using R-GMA APIs. In that way Grid security will be also implemented. And the fact that the developed web services can be used in more complex processes – sequences of related tasks, can be used for specification, development and deployment of business processes in high-energy physics.

IBM provides service-oriented framework for business process specification. There are tool that realized this framework and provides very good integration with WSRR and Data Studio Developer tools, that we used in our research. However business process specification, development and deploy of high-energy physics is a subject of future investigation.

Acknowledgements

This work was supported by University of Sofia SRF under Contract N163/2010.

References

1. Foster, I., Kesselman, C., The Grid: Blueprint for a New Computing Infrastructure - Second Edition, Morgan Kaufmann Publishers, 2004.
2. Programming the grid with g-Lite, <http://cdsweb.cern.ch/record/936685/files/egee-tr-2006-001.pdf>
3. R-GMA User Guide for Java Programmers, <https://edms.cern.ch/file/503617/6.0.0/EGEE-JRA1-TEC-503617-v6.0.pdf>
4. Apel User Guide, <http://www.egee.cesga.es/EGEE-SA1-SWE/accounting/guides/apel-user-guide-glite.pdf>
5. IBM Data Studio Data Web Services, Part 1: IBM Data Studio: Get started with Data Web Services,
<http://www.ibm.com/developerworks/data/tutorials/dm0711pauser/dm0711pauser-pdf.pdf>
6. Data migration and change management using Data Studio Administrator V2.1
<http://download.boulder.ibm.com/ibmdl/pub/software/dw/data/dm-0904changemgmt/dm-0904changemgmt-pdf.pdf>
7. WebSphere Service Registry and Repository Handbook,
<http://www.redbooks.ibm.com/redbooks/pdfs/sg247386.pdf>

Controlling presentation slides with tangible passive markers

Svetoslav Neykov

Sofia University „St. Kliment Ohridski“
sneykov@fmi.uni-sofia.bg

Abstract. This paper describes the usage of printed markers as an effective means of interacting with a presentation. By mapping to existing real world behaviour the process of controlling the slides is made more natural. In addition to changing slides means for more complex actions such as zooming/panning, executing embedded actions are presented.

Keywords: tangible markers, human computer interaction

*This work is supported by Project VU-MI-204/2006
“Intelligent sensor systems for security enhancement”
funded by the Bulgarian National Science Fund.*

1 Introduction

Historically the mouse and keyboard have instituted as the most practical alternatives for communicating our intents to a computer. They force us to accommodate to a specific usage paradigm instead of the other way around. With the rise of pervasive computing this is becoming increasingly obvious.

Commercial grade success has been reached by gesture based interfaces but this is restricted to surface contact only (touchpads, touch sensitive screens). Reliable general purpose gesture recognition, voice recognition and alternative approaches like eye tracking are yet to be developed.

Even though the keyboard will be the main alternative for general purpose high throughput human-computer interface for a long time to come we are seeing increasing development of strongly specialised interfaces for predefined tasks. Such interfaces should augment the user behaviour instead of making him get out of his way.



2 Related Work

Using markers in computer vision applications allows for real-time robust tracking of an object and its pose relative to the camera. The ARToolkit[1] framework allows for registering custom images as markers. More recently developed frameworks such as ARTag[2] use digital markers (similar to 2D bar-codes) which improves the reliability and speed of registration. Building on the existing work ARToolkitPlus[3] improves the digital marker by introducing a checksum for improved registration.

Two usage patterns of fiducial markers can be distinguished. One is simply overlaying virtual objects over the marker, making them look natural in the physical world recorded by the camera. Another is interacting with the application by manipulating the marker. Video conferencing solutions use them as virtual monitors to project the participants' videos[4]. Remote collaboration environments employ both approaches by using dedicated markers for introducing objects into the system and acting on those objects.

Fiducial markers prove useful outside of the augmented reality field as well. They can be used both as general purpose input device, mimicking a mouse [5] as well as specialised for a concrete task such as playing music[6].

An alternative approach to controlling presentation slides has been proposed by [7]. By rotating the marker clockwise and counter clockwise the slides advance and reverse correspondingly. The main advantage over traditional presentation remote controls is the low cost and availability.

2 Choosing an Appropriate Marker

The objective of this work is to design an intuitive method for controlling presentation slides using physical cues. The idea of using presentation note cards is extended to pair each card with the slide that should be presented. This is accomplished by assigning separate markers to each slide, printing them on the back of card-sized pieces of paper and sorting them in a deck. The front side of the card can be used for notes regarding the slide, scaled image of the slide or any other content the user finds appropriate.

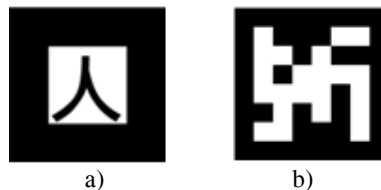


Fig.1. Markers used in the tests.

Several types of markers have been considered. Tests were performed with printed markers of size 10 per 10 centimeters held at one meter from the camera without prior calibration. The simplest markers (Fig. 1a) as used by ARToolkit are custom images in a black square. In the performed tests the registration of the marker strongly

depended on the image and behaved poorly at the chosen distance. The digital markers (Fig. 1b) introduced in ARTag significantly improve the registration of the markers to the point of being usable for the considered task. The markers that were chosen as the most appropriate are the ones implemented by ARToolkitPlus. They build on the ARTag digital markers and introduce a checksum which helps for reconstructing the image in non-ideal lightning conditions.

The image stream from the camera is constantly scanned for markers. The detection used in ARToolkitPlus takes place in several steps. The image captured from the camera is converted to greyscale (Fig. 2a) and is thresholded (Fig. 2b). The threshold is fixed but if no marker is found the threshold is repeated with a new randomly chosen level. The current implementation uses maximum of two iterations. At the next step contours consisting of four vertices are enumerated (Fig. 2c). After normalizing the regions the sub-images are checked for valid patterns.

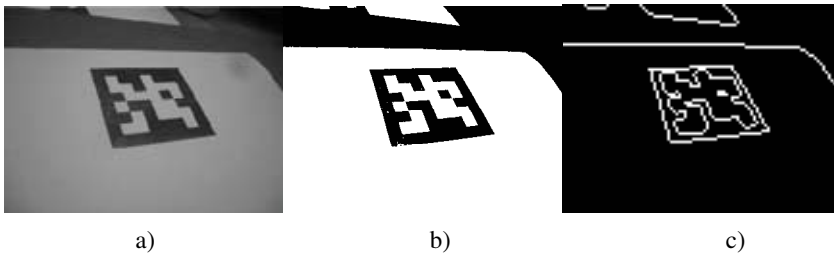


Fig.2. Marker registration process.

3 Specialised Interface for Controlling Presentations

To switch to a particular slide the user just exposes the back of the card to the camera. In this way each slide is physically correlated to the card with its corresponding marker. There is no specific forward/backward action. The system assumes that no two markers will be visible at the same time and if such condition happens, the latest to appear will take precedence.

The user receives tactile feedback of his position in the presentation by the thickness of the card deck. Seeking for a particular slide is facilitated as it is easier to search through the deck of cards in hand than to flip the slides. It is no longer needed to scroll sequentially through the presentation.

Extended actions are possible besides changing the slides. To activate the manipulation mode the marker should be rotated sideways. Once in that mode the slide can be zoomed by approaching the card to the camera and panning by moving it sideways. The original state will be reset once the normal orientation of the marker is restored or it disappears from the camera view. If the marker is hidden for a short period of time (1-2 seconds) in the sideways orientation an embedded slide action is activated (corresponds to mouse click).

Although more involving than a simple one-click remote control the proposed interface provides natural mapping to existing paradigms from the physical world.

4 Conclusion

A specialised interface for interacting with presentation slides using fiducial cues was introduced. Using tangible markers as an input interface to a computer brings a more natural experience to the user, giving him physical and real-time tactile feedback. With the advance of ubiquitous computing such interfaces are becoming increasingly useful for situations where no access to conventional input means is practical. Future work will investigate the registration precision of more user friendly markers such as plain numbers. To improve the usage in poor lighting conditions adaptive thresholding methods will be compared in both accuracy and processing speed.

References

1. Kato, H., Billinghurst, M. (1999) Marker Tracking and HMD Calibration for a video-based Augmented Reality Conferencing System. In Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99).
2. Mark Fiala, ARTag, a Fiducial Marker System Using Digital Techniques, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, p.590-596, June 20-26, 2005
3. Wagner Daniel, Schmalstieg Dieter, ARToolKitPlus for Pose Tracking on Mobile Devices. Proceedings of 12th Computer Vision Winter Workshop (CVWW'07) February 2007
4. Kato, H., Billinghurst, M., Morinaga, K., Tachibana, K. (2001) The Effect of Spatial Cues in Augmented Reality Video Conferencing. In proceedings of the 9th International Conference on Human-Computer Interaction (HCI International 2001), August 5-10th.
5. E. Woods, P. Mason, M. Billinghurst. MagicMouse: an Inexpensive 6-Degree-of-Freedom Mouse. Proceedings of Graphite 2003, Feb 11th-13th, 2003, Melbourne.
6. E. Costanza (2003) D-touch: A consumer-grade tangible interface module and musical applications. Proceedings of Conference on HumanComputer Interaction (HCI03), p. 8-12
7. Crease, M. Fiala, M. (2007) Presentation Assistant and Kiosk Interaction with Fiducial Markers. The IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE'2007)

Hello World in Parallel!

An introductory lab exercises in parallel/concurrent and distributed programming

Hristo Hristov¹

¹Faculty of Mathematics and Informatics, St. Kl. Ohridski University of Sofia, Bulgaria

Abstract. This article describes the design and structure of introductory course in parallel/distributed programming taught in Faculty of Mathematics and Informatics at Sofia University. We describe motivation for the design of this course and selected platform. We also outline material included in different lab exercises in the course.

Key words: Parallel programming, Parallel programming course, University programming course

1 Introduction

In modern information technology world multi core computer systems are beginning to play extremely major role. Almost every new computer system these days comes with multi core central processor unit (CPU). Even low budget home user systems are with at least two CPU cores, and when it comes to high-end workstations or servers, multi core processors are mandatory standard.

Being so affordable, these new computer systems raise a number of questions when it comes to software run on them. At first place software should be designed as “multi core” aware. It should understand and exploit at maximum level multi core architecture. At second place, supporting and developing such application is not very much like traditional serial programming, so new developer abilities are needed.

And last, but not least, every modern operation system comes with complex graphical user interface which in turn requires program to respond simultaneously on number of different events. One major part of them are raised from operating system and others are as a result of user activities.

When we are talking about desktop computing, the list of specified questions may look sufficient. I.e. when a given programmer has skills in a certain language and selected application framework (in terms of their “beloved” operating system), everything fits perfectly. He can design and develop “multi core” applications and if it is a good framework then his program will use at maximum level computer architecture.

Even mentioning only cross-platform computing, server side programming, grid or cloud computing we define a new level of requirements to software developers and their skills. Terms like operating system process management, network programming and interprocess communication are playing a major role in this kind of programming.



The main purpose of the course was to introduce students with the very basics of these concepts. We tried to select well known programming language and platform, which covers well both types of programming – threaded/concurrent and parallel. Besides this, cross-platform program development was also an issue in choosing programming language for the course.

2 Course pre-requirements

When a new to the subject developer meets parallel world a number of questions arise. It looks like those proficiency skills at certain programming language are not sufficient for writing good parallel programs. He or she should be familiar with concepts which are subject on other university courses, like “Computer Architectures”, “Computer Networks” and “Operating Systems”. Among these concepts, depending on the listed subjects are:

- a) Computer Architectures:
 - machine organization – bits, bytes, words, double words;
 - central processor unit;
 - memory organization;
 - CPU context/task switching;
- b) Computer Networks:
 - packet, protocol;
 - host, IP address, protocol – URI (universal resource identifier);
- c) Operating Systems (POSIX function names are used):
 - process/light process (threads) – fork()/exec(), pthread_create()/pthread_exit();
 - sockets – bind(), listen(), accept(), connect();
 - interprocess communications.

Having this in mind, parallel world fast becomes a bit complex for a developer with good understanding in particular programming language, but little or none of the listed terms. From this point of view course programming language and selected platform should satisfy a number of requirements:

- well known (industry) programming language with low learning overhead;
- easy to install development environment;
- platform independence (minimal hardware platform dependence);
- clear presentations of listed pre-requirements (even better if modeled with OO models) ;
- integrated “distributed” capabilities;
- modeling of process/light process with native programming language statements/libraries (without need of knowledge in operating system level programming).

Taking into account language requirements, we decided to use Java as programming language and Java SE SDK as platform distribution [1]. Versions 1.5.x and 1.6.x were used.

A feature rich and industry standard for Java development - Eclipse [2] software package was used as integrated development environment. We used standard version - Eclipse IDE for Java Developers.

3 What does Java give us?

Java is not only a programming language. During the recent years it has evolved into an industry standard. Nowadays we refer to Java, not only as a language but more as a software platform. It has strong and clear object-oriented models for a variety of industry standards. Moreover it is a platform-independent, rich and free environment. In this point we will list some of Java language (and Java platform) features which were used in our introductory course.

We started lab exercises with Java thread model. Both `java.lang.Runnable` interface and `java.lang.Thread` container were covered. Our first aim was to introduce students to lightweight process creation and termination. Thread states (New, Runnable, Blocked, Waiting, Timed Waiting, Terminated) were also well addressed [3], [5].

At second place we introduced Java network concepts [4]. Both `ServerSocket` and `Socket` were introduced. Terms host, IP address and port were used and viewed in practice. Combining Java Threads and Java Networking we met network programming world and a very beginning of server side programming.

Considering thread and network functionality in Java we introduced the concept of single resource sharing and problem of thread synchronization. Race condition examples were shown and analyzed. Introducing lock mechanisms in Java to handle the problem, we also referred another bottle neck of concurrent programming - dead locks [5].

Our next step was to move from single machine level to distributed one. Java RMI was introduced [6]. Students met Java remote object presentation. We developed an access to remote resource examples, using native Java code.

Combining RMI and Threads we achieved fully asynchronous request handling. Also, using RMI, a concept of remote side call back functionality was shown.

For all of listed Java concepts we developed and run an example source code [7]. Next step was to introduce concurrent/parallel programming world.

4 Course teaching, example parallel algorithms and feedback

After we introduced students to particular concurrent/parallel “tool set” of certain programming language our last aim was to show them how to apply the learned material in real world parallel programs. “From serial code to parallel code”-approach was used. Given a well known example serial algorithm we aimed ourselves at making it parallel. Excellent “practical” approach in [8] was used – considering a

working serial code we discussed different techniques to make it parallel. One of them was used in parallelized version of the code. We did not try to formalize the exposure of parallelized version, but rather to reveal “magic” that makes them parallel.

Both task decomposition and data decomposition approaches to parallelization were introduced. If our example problem can be divided in different tasks then we use task decomposition approach to divide tasks among concurrent/parallel threads. Other technique - data decomposition illustrates how to divide the input data among different tasks (threads). Every example was designed to show overhead (time spent in system calls) because of parallelization and real computation work. For a given task size we measured achieved speedup depending on number of cores the task was executed.

We will list a brief description of considered example projects:

- Parallel sum – a pure data decomposition algorithm with simplest ever serial version. Algorithm finds a sum of one dimensional array of integer elements. Parallel version did not use any specific thread management.
- Prefix scan – partial sum of array elements. Both data decomposition and task decomposition approaches were used in this example. Main purpose was to show that if we do not terminate task threads but synchronize them we can lower the overhead during task parallelization;
- Parallel Bubble – an exotic parallelization example, showing another way of multiple threads working on common resource – wavefront approach [8]. Qualification “wavefront” comes from the way we run threads on common data set – their passing through data recalls the way sea waves run to coast. Although this algorithm has a complex synchronization part, it shows up to 70% speed up on certain task sizes (despite such a slow algorithm like Bubble Sort);
- Parallel Depth First Search – Main purpose of this example was to show that not every serial algorithm can be easily made parallel. The graph used in the example was randomly generated and presented with adjacency matrix.

We finished our introductory course with personal projects, based on considered examples. Projects were assigned in last two weeks of the term. Students had 3 weeks in total to complete and present their projects. All example code and project tasks can be accessed at [7].

Course was organized in lectures (3 hours per week) and lab exercises (2 hours per week). Lab exercises took place in computer rooms with 20 seats and standard desktop computers under Microsoft Windows operating system.

A great number of students showed interest in the subject, although there was a lot of “cross-subject” material involved in the course. We received requests for extending course material in two directions. One was to show more Java “parallel” features, and another was to include more parallel algorithms examples in the lab exercises.

5 Conclusion and future steps

Although it was an introductory course a great number of students showed interest in developing a personal project. Almost all (90%) of them succeeded in achieving and showing a speedup with their parallel version of selected algorithm.

As our experience with this course has shown there is a big interest in concurrent/parallel programming. More over modern hardware and operating systems demands it. For future tasks we state to expand the course with more parallel programming examples and more interesting personal projects. New hardware architectures like multi core cell processors are beginning to play major role in modern computing and their program model can be also addressed like a normal prolongation to this course.

Acknowledgements: The paper is supported by Grant 162/2010 from Sofia University Research Fund.

References

1. Oracle/Sun distribution website: <http://java.sun.com/javase/downloads/index.jsp>
2. Eclipse website: <http://www.eclipse.org/downloads/>
3. Horstmann, C. S., Cornell G.: Core Java, Volume I – Fundamentals (8th edition)
4. Horstmann, C. S., Cornell, G.: Core Java, Volume II – Advanced Features (8th edition)
5. Oracle/Sun Concurrency tutorial:
http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/essential/concurrency/
6. Oracle/Sun RMI tutorial:
http://download.oracle.com/docs/cd/E17409_01/javase/tutorial/rmi/index.html
7. Course examples source code: <http://rmi.yaht.net>
8. Breshears, C.: The Art of Concurrency - A Thread Monkey's Guide to Writing Parallel Applications

E-commerce Payment Systems

Ana Serafimova

Abstract. From the times when people exchanged seashells for fish through the use of paper money, merchants have always tried to simplify the payment process. Nowadays, thanks to the growth of the Internet and e-commerce, payment is as easy as sending an email. More and more people prefer the convenience of sending or receiving money via the various available methods for electronic payment. The main purpose of this article is to present an overview of some of the currently used systems for conducting e-commerce transactions.

Keywords: electronic payment systems, e-commerce, e-commerce payment systems, credit cards, electronic money, digital wallets, mobile payment, Google Checkout, AlertPay, PayPal, Moneybookers, payment service provider, payment processor, online payment processing system

1 Introduction

The earliest form of payment was the exchange of goods and services for other goods and services. Later people started using various kinds of money, the earliest of which was called commodity money [1]. “Commodity money is money whose value comes from a commodity out of which it is made” (such as large stones, salt, copper or gold) [2]. The most commonly used commodity money were gold and silver coins since they were portable and divisible [1].

The next stage in the history of money was the use of paper notes which corresponded to deposits of gold and silver owned by the note issuer.

Today nearly all modern money systems are based on fiat money. “Fiat money is without value as a physical commodity, and derives its value by being declared by a government” [3] to be “a form of payment within the boundaries of the country” [3].

1.1 The Internet and the Age of E-commerce

Nowadays, more and more people are using electronic methods for payment instead of sending checks through the mail or handling any paper money. The idea of purchasing goods and services electronically is not a new one. Since the late 1970s and early 1980s, many ways have been proposed to enable payment to be accomplished across a computer network. But they were of little use to those who were not connected to a network [1].

The appearance of the Internet has changed this and today about 2 billion people have access to the Web [4].



This network of networks became the catalyst for the revolution called electronic commerce and electronic business. But this new type of business required new payment methods, different from the traditional payment systems that could fulfill its special needs. Thus various types of electronic payment systems were created to support further development of e-commerce.

According to a “global survey conducted by The Nielsen Company, over 85 percent of the world’s online population has used the Internet to make a purchase” [5]. The Nielsen survey was “conducted from October to November 2007 and polled 26, 312 Internet users in 48 markets in Europe, Asia Pacific, North America and the Middle east” [5]. The highest percentage of Internet users who have shopped online was found in South Korea (99%), Germany (97%), Japan (97%), UK (97%), US (94%) [5]. The results of the same survey show that the most popular items purchased online were books, clothing/accessories/shoes, videos/DVDs/games, airline tickets and electronic equipment [5].

Initially in e-commerce, the Web was used as a way of searching for products and services and the payment was done off-line using traditional payment methods.

However, within the past ten years, e-commerce has undergone explosive development that resulted in the emergence of many types of electronic payment methods currently available to network users. “In its most general form, the term electronic payment includes any payment to businesses, bank or public services from citizens or businesses, which are executed through a telecommunications or electronic network using modern technology” [6].”Furthermore, the payment is made from distance, without the physical presence of the payer and naturally it does not include cash” [6].

This article describes the characteristics of some of the more popular electronic payment systems used in e-commerce for the acceptance of payment over the Internet. These include traditional methods such as credit cards and also new technologies such as digital wallets, electronic money and mobile payment. The article also presents a review of some of the most popular payment service providers (also known as online payment processors). They represent another form of electronic payment that allows the online transactions to be completed by a third party [7].

2 Review of Some of the Currently Used Electronic Payment Methods

The next few sections present a description of the following electronic payment systems: credit cards, digital wallets, electronic money, mobile payment and online payment processing systems.

2.1 Digital Wallets

Digital Wallets Technology. This section is compiled on the base of sources [8, 9].

A digital wallet is a software component that provides information such as shipping address and credit card number for use in e-commerce transactions. Digital

wallets are quite similar to physical wallets and they allow users to order and purchase goods and services online easily and securely without the need to enter sensitive information that is vulnerable to theft by hackers [9].

A digital wallet consists of two parts – a software and information component. The software component is responsible for the required security and encryption of the available personal information. The information part is a database used for storage of information used in Internet transactions such as shipping address, billing address, credit card numbers and other information [9].

Digital wallets can be stored either on the client side or on the server side. Client-side digital wallets are easily maintained by the customers themselves while server-side digital wallets are created by an organization such as a digital wallet vendor or Internet service provider and maintained on their servers [9].

Making a purchase online requires the filling of a form which can be a long process. Digital wallets automatically fill out the order form with the required information, including customer's name, credit card number, billing and shipping information, and then send the information to the online merchant. In addition, after entering the required information, the digital wallet also encrypts the data [8].

2.2 Electronic Money

This section is compiled on the base of sources [10, 11].

Definition of Electronic Money. Electronic money (also known as e-money, e-cash, digital money) is the electronic alternative to cash used for payment in online transactions. "Technically electronic or digital money is a representation, or a system of debits and credits, used to exchange value, within another system, or itself as a stand alone system, online or offline" [12]. E-money can be stored on cards and devices or on a server, for example prepaid credit cards¹, electronic purses² or web-based services, such as PayPal.

Compared to ordinary paper cash, e-money costs less in a transaction. When transferring electronic money the transaction cost does not depend on the amount. In addition, the usage of e-cash is anonymous and does not require any contacts with a bank or the filling of documents [30].

Types of Electronic Money. There are two types of e-money: identified e-money and anonymous e-money. Identified e-money contains information about the person who withdraws money from the bank. This kind of e-cash can be tracked by the bank as the money moves through the economy. Anonymous e-money is very similar to real paper money since it can be used in transactions without leaving any trails [10].

¹ A prepaid credit card is "a payment card (usually MasterCard, Maestro, Visa Electron or Amex), pre-loaded with your own money, which you can then use wherever the payment card is accepted, including on the internet and abroad" [13].

² An electronic purse is a "type of smart card which, with an embedded microchip, provides multiple options, such as debit card or credit card type payments" [15].

Each of the mentioned types of e-money has an online and an offline variant. Online e-money require interaction with a bank (through a network) in order to be used in a transaction with a third party, while the usage of offline e-money does not involve a bank [10].

2.3 Mobile Payment

This section is based on source [14].

Since the emergence of mobile phones they have been equipped with a lot of functionalities which have supported the development of mobile services and mobile commerce and offer very profitable opportunities to merchants and service providers.

Mobile payment services started gaining popularity in late 1990s and early 2000s [14]. Unfortunately, many of the efforts in the sphere of mobile payments failed possibly due to the fact that mobile technologies were still not mature enough and difficult to use and therefore could not attract customers, merchants and banks. Today this has changed with all the innovations added to the mobile technologies.

“Mobile payments are payments for goods, services, and bills/invoices with a mobile device (such as a mobile phone, smart-phone, or Personal Digital Assistant) by taking advantage of wireless and other communication technologies (such as mobile telecommunications networks, or proximity technologies)” [14]. Mobile payment allows users to make purchases such as news, music, games, concert or flight tickets, parking fees or pay bills and invoices through electronic payment services. Making a mobile payment requires connection to a server through the mobile device in order to perform authentication and authorization as well as notification of the completed transaction.

2.4 Credit Card Payment Systems

This section is compiled on the base of sources [16, 17].

Credit Cards Definition. According to Wikipedia “a credit card is part of a system of payments named after the small plastic card issued to users of the system” [16]. It is a card containing identification information such as a signature or picture that allows the user “to charge purchases or services to his account – charges for which he will be billed periodically” [17]. The information on the credit card can be obtained from a magnetic stripe or chip on the card by bank computers, automated teller machines (ATMs) or store readers.

Credit cards were first used in the United States during the 1920s when companies, such as hotel chains, started issuing them to their customers [17]. In 1950 the company Diners Club, Inc. introduced the first universal credit card which could be used at different stores and businesses [17].

How Credit Cards Work. Credit cards are issued by a credit provider after the approval of an account by the same provider. After that the cardholder can use this card to make purchases and whenever a purchase is made, the user has to pay the card

issuer. This is done by signing a receipt containing details about the card and the amount to be paid or by entering a PIN (personal identification number) [16].

The validation of the card is done through the use of electronic verification systems which allow merchants to verify that the customer has sufficient funds to cover the purchase they are making. These systems include credit card payment terminals or Point of Sale systems (POS) connected to the merchant's acquiring bank³.

Every month the cardholder receives a statement containing information about the purchases made with the card and the total amount that the user owes. After receiving the statement, the user must pay a minimum part of the bill by a due date or pay a greater amount [16].

2.5 Online Payment Service Providers

This section is based on sources [18, 31].

Online payment service providers are systems that provide processing of payments in online transactions between customers, businesses and banking institutions. According to Wikipedia a payment service provider (PSP) offers merchants online services for accepting electronic payments by a variety of payment methods including credit card, bank-based payments such as direct debit, bank transfer, and real-time bank transfer based on online banking [18]. Online payment processing systems are software applications used to facilitate e-commerce transactions but they can also be used for processing payments in traditional brick and mortar businesses. In addition, they provide encryption of personal data, authorization of payments and communication between the customer and the participating businesses and financial institutions.

Usually, the processing of an online transaction by a payment service provider includes the following steps [31]:

1. The customer makes a purchase (for example via a web site or phone).
2. The information that is sent is "scrambled" by the Internet browser (using Secure Socket Layer (SSL) encryption) used by the customer.
3. The information entered by the user is sent by the web site to the payment gateway.
4. The payment service provider sends the received information to the appropriate bank.
5. The request is forwarded by the bank to the card association. If the bank is the same as the card association (such as American Express or Discover) then a direct response can be delivered. Otherwise, for example in the case of MasterCard or Visa, there are some extra steps that need to be undertaken:
6. The card association sends the information to the bank that issued the card (the customer's bank).
7. The customer's bank determines whether the customer's funds are enough to pay for the purchase.

³ An acquirer (or acquiring bank) is a member of a Card Association, for example MasterCard and/or Visa, which maintains merchant relationships and receives all bankcard transactions from the merchant [19].

8. The bank sends an authorization code and depending on that the transaction may be declined or accepted.
9. The authorization code is sent by the payment processor to the payment gateway.
10. The payment system sends the code to the business web site and depending on the code the sale is allowed or denied.

The next few sections provide an overview of the main features of some of the currently available online payment service providers.

Google Checkout. This section is compiled on the base of the sources [20, 21, 22].

Google Checkout is an electronic payment system provided by Google that started functioning on June 28th, 2006 [20] and is growing in popularity since then. In the year 2008, the Google Checkout payment processing system had a market share of 11% [20]. The system is easy to use and simplifies the process of making online purchases. Users can use a credit card or debit card to pay for their purchases. The credit or debit card and shipping information is stored in the user's Google Account. Users of Google Checkout do not need to pay for set up or any monthly fees. There is only a transaction fee that is much lower compared to many other payment gateways. The fee depends on how much money is used in sales [21]:

1. Less than \$3,000 per month: 2.9% of each transaction, plus \$0.30.
2. \$3,000 to \$9,999.99 per month: 2.5% of each transaction, plus \$0.30.
3. \$10,000 to \$99,999.99 per month: 2.2% of each transaction, plus \$0.30.
4. More than \$100,000 per month: 1.9% of each transaction, plus \$0.30.

Google Checkout is accepted by most major e-commerce web sites.

An important feature of Google Checkout is the great fraud protection provided by the system. The users can be sure that their information will not be revealed to the merchant.

However, Google Checkout has a few disadvantages and one of them is the fact that the system does not support international payment forms which means, for example, that a customer cannot spend Euros to make purchases using Google Checkout. This could lead to loss of customers. Another disadvantage is that Google checkout is not accepted by eBay and this could be another reason for customers to be unwilling to use Google Checkout [22].

Google Checkout is a good choice for small businesses that cannot afford the services of other more expensive payment processors.

AlertPay. This section is based on source [23].

AlertPay is a payment processing system with free registration that allows users to send or receive money via Internet. The company was established in Canada in June 19th, 2005 [23]. Users can send or receive payments from other users of AlertPay in which case both parties receive a confirmation email for the payment. It is also possible to send money to non-members of AlertPay. In that case the non-member is

notified that they have received money via AlertPay and they are required to create an AlertPay account.

Members of AlertPay are able to purchase products and services online. The user is first redirected to the AlertPay secure server to complete the order process and both parties receive a confirmation email for the payment.

There is no fee charged for sending money, but when the user receives money there is a fee depending on the type of the account. There are three types of account [23]:

1. Personal account: The user is allowed to receive a maximum of \$400 per month and \$2000 per year without paying a fee.
2. Premium account: There is no limit for the received amount of money. A premium member has access to additional features such as Mass Pay and website payment methods. The fee charged for receiving money is 2.5% + \$0.25 per transaction.
3. Secured account: Provides the same features as the premium account and additional security. The fee for receiving money is 3.9% + \$0.59 per transaction.

Moneybookers. This section is compiled on the base of sources [24, 25, 26].

Moneybookers is an online payment processor that enables customers to send and receive money using an email. It is possible to send money from a credit/debit card or transfer money to and from one's bank account. Moneybookers was founded in 2001 and is owned by Investcorp Technology Partners. The service is one of Europe's largest online payment systems having over five million users [24], the majority being from UK and Europe.

Before getting access to the system, users need to provide identity verification documents. Once the verification is approved, the user is allowed to send money to anyone with a valid email address. If the recipient is not registered in Moneybookers, he/she need to create a Moneybookers account (which is free) in order to receive the money.

The most common methods for uploading money to a Moneybookers account are via bank transfer or credit card. If bank transfer is used there is no additional fee but if the uploading of money is done through a credit card, the customer is required to pay a fee that is 1.90% of the total amount [25].

Members of Moneybookers are able to make a variety of online purchases (including online gaming, MMORG's) at more than 200000 online shops [26]. The system is not only a convenient way for conducting online transactions but it is also safe. Users can be sure that their personal and financial information is not revealed to online businesses.

PayPal. This section is compiled on the base of sources [27, 28, 29].

"PayPal is an e-commerce business allowing payments and money transfers to be made through the Internet" [27] and it "serves as an electronic alternative to

traditional paper methods such as checks and money orders” [27]. The system allows users to buy and sell products and services online safely and easily. PayPal started working in 1999 as a means of sending and receiving payment in eBay auctions [28]. Today it is the world’s most popular online payment system with over 99 million users.

Members of PayPal are allowed to send and receive payments through the system and the information that PayPal needs in order to transfer the money is a valid email address and a valid credit card or bank account.

Members of PayPal do not need to pay for sending money to other PayPal users. The money is withdrawn from the user’s credit card or bank account. When receiving money though, sellers are charged with a fee of 1.9% to 2.9%. The following table shows the fee for the corresponding amount of received money [29]:

Table 1. Amount of received money and their corresponding fees.

Received transfer	Charged fee
Under \$3000	2.9% + \$0.30
\$3000.01 to \$10,000	2.5% + \$0.30
\$10,000.01 to \$100,000	2.2% + \$0.30
Over \$100,000	1.9% + \$0.30

PayPal is owned by the company eBay so buyers and sellers are advised to set up a PayPal account. Over 90% of the merchants participating in eBay use PayPal as a method for receiving payments.

PayPal is a convenient and safe way for transferring money between its members using email addresses. It is also possible to send money to anyone even if they do not own a PayPal account – in this case recipients will be required to set up an account for free.

When transferring money or sending payments, users can be sure that their credit card and personal information will be kept secret and will never be revealed to the other party. The only information that is made public to the seller is the customer’s email and shipping address. It could be said that among the available online payment systems PayPal provides the best security to its members and protects them from fraud and identity theft.

When setting up a PayPal account, users are asked to provide the following information:

- A valid email address.
- A valid credit card and/or a valid bank account.

PayPal offers three different types of account:

1. Personal account: This type of account is free and allows users to send and receive money. No fee is charged for conducting transactions via PayPal. However, users are allowed to receive payments only from other PayPal accounts but they cannot accept money from credit or debit cards.
2. Premier account: To set up a premier account customers need to pay a certain fee. This type of account offers additional features such as receiving

funds from credit or debit cards and a “shopping cart”. Members are able to upgrade from a Personal to a Premier account at any time.

3. Business account: Suitable for large online businesses or online shops. A business account provides features that make it easier for merchants to manage large amounts of sales. Owners of this type of account are required to pay a fee.

When a user sends money PayPal withdraws the funds from the user’s credit card or bank account and sends them to the recipient within two business days. Both parties are notified via email that the money has been sent/ received. When a member receives money PayPal sends it to the user’s bank account via wire transferring. This procedure may take up to eight business days.

3 Conclusion

The Internet has revealed a new world of shopping through various methods for online payment. Every customer or merchant chooses a method that best suits their needs based on its specific features, advantages and disadvantages. According to the Nielsen survey conducted in 2007, credit cards are the most common method of payment for making online purchases.

Online payment processors are a convenient and safe method for sending and receiving online payment. In the aforementioned Nielsen survey PayPal was rated as the second most common online payment method and it’s the most popular system of its kind.

References

1. O’Mahony, D., Peirce, M. and Tewari, H.: Electronic payment systems for e-commerce. London: Artech House (2001)
2. Wikipedia article on commodity money, http://en.wikipedia.org/wiki/Commodity_money, last visited 11. 03. 2010.
3. Wikipedia article on fiat money, http://en.wikipedia.org/wiki/Money#Fiat_money, last visited 11. 03. 2010.
4. Internet usage statistics: World internet users and population statistics, <http://www.internetworldstats.com/stats.htm>, last visited 11. 03. 2010.
5. Nielsen: Trends in online shopping. A global Nielsen consumer report (2008)
6. Sumanjeet, S.: Emergence of payment systems in the age of electronic commerce: The state of art. Global journal of international business research, vol. 2, num. 1 (2009)
7. Wikipedia article on e-commerce payment systems, http://en.wikipedia.org/wiki/E-commerce_payment_system, last visited 11. 03. 2010.
8. Gale encyclopedia of e-commerce: Digital Wallet Tecnology, <http://www.encyclopedia.com/doc/1G2-3405300137.html>, last visited 11. 03. 2010.
9. Wikipedia article on digital wallets, http://en.wikipedia.org/wiki/Digital_wallet, last visited 11. 03. 2010.

10. Miller, J.: Answers to frequently asked questions about electronic money and Digital Cash, <http://projects.exeter.ac.uk/RDavies/arian/emoneyfaq.html>, last visited 11. 03. 2010.
11. E-money – mobile money – mobile banking – what’s the difference (2009), <http://psdblog.worldbank.org/psdblog/2009/01/emoney-mobile-money-mobile-banking-whats-the-difference.html>, last visited 11. 03. 2010.
12. Wikipedia article on electronic money, http://en.wikipedia.org/wiki/Electronic_money, last visited 11. 03. 2010.
13. What is a prepaid card?, <http://www.what-prepaid-card.co.uk/features/what-is-prepaid.html>, last visited 11. 03. 2010.
14. Dahlberg, T., Mallat, N., Ondrus, J. and Zmijewska, A.: Mobile payment market and research – past, present and future.
15. Definition of electronic purse, <http://www.businessdictionary.com/definition/electronic-purse.html>, last visited 11. 03. 2010.
16. Wikipedia article on credit cards http://en.wikipedia.org/wiki/Credit_card, last visited 11. 03. 2010.
17. How credit cards work, <http://money.howstuffworks.com/personal-finance/debt-management/credit-card.htm>, last visited 11. 03. 2010.
18. Wikipedia article on payment service providers, http://en.wikipedia.org/wiki/Payment_service_provider, last visited 11. 03. 2010.
19. Wikipedia article on acquirer, <http://en.wikipedia.org/wiki/Acquirer>, last visited 11. 03. 2010.
20. A review of Google Checkout – cheaper, faster, safer, easier than its competitors, <http://www.examiner.com/x-312-Auctions-and-Antiques-Examiner~y2009m3d3-A-review-of-Google-Checkout--online-merchants-are-starting-to-find-it>, last visited 11. 03. 2010.
21. Google Checkout Review, <http://www.bestpaymentgateways.com/articles11.html>, last visited 11. 03. 2010.
22. Google Checkout Review, <http://blog.100best-merchant-accounts.com/2009/09/google-checkout-review.html>, last visited 11. 03. 2010.
23. AlertPay Review, <http://www.squidoo.com/alertpay>, last visited 11. 03. 2010.
24. Moneybookers Review, <http://www.howtopayonline.org/moneybookers.html>, last visited 11. 03. 2010.
25. Moneybookers Review, <http://www.pokernews.com/deposit-options/moneybookers/>, last visited 11. 03. 2010.
26. Moneybookers review, http://www.letscomparebets.com/fixed_odds_bookmakers/moneybookers.html, last visited 11. 03. 2010.
27. Wikipedia article on PayPal, <http://en.wikipedia.org/wiki/PayPal>, last visited 11. 03. 2010.
28. Wilson, R.: PayPal review (2001), <http://www.wilsonweb.com/reviews/paypal.htm>, last visited 11. 03. 2010.
29. Gil, P.: PayPal 101: How PayPal works, <http://netforbeginners.about.com/od/ebay101/ss/paypal101.htm>, last visited 11. 03. 2010.
30. What is electronic money?, http://www.e-money.com/articles/4_stat.html, last visited 11. 03. 2010.
31. What is a payment gateway?, <http://www.bestpaymentgateways.com/articles3.html>, last visited 11. 03. 2010.

Popular Genealogy Software Products

Darina Serafimova

Abstract. The study of families and the tracing of their lineages and history have been important since ancient times. Nowadays, this study is known as genealogy and is popular not only as a profession, but also as a hobby. Both the professional and hobbyist genealogists could make use of genealogy software products in order to facilitate their work. The main purpose of this paper is to present an overview of some of the most popular genealogy software products. Before this overview, an introduction to the field of genealogy is presented, as well as a review of the history of the genealogy software.

Keywords: genealogy, genealogy software, Family Tree Maker, Legacy, RootsMagic, The Master Genealogist

1 Introduction

Before presenting the overview of the selected genealogy software products, some general information about genealogy and genealogy software products should be given. This information is necessary in order to be understood how the genealogy software has come into existence and why is it useful for genealogists.

The introduction is divided into two sections. In the first section, entitled “Genealogy Overview”, an overview of the field of genealogy and its history is presented. In the second section, entitled “Genealogy Software Overview”, an overview of genealogy software products and their history can be found.

1.1 Genealogy Overview

Encyclopedia Britannica [1] and Wikipedia [2] define genealogy as “the study of family origins and history” and “the study of families and the tracing of their lineages and history”, respectively. Genealogists use various data sources, including verbal, written, genetic analyses, etc., in order to acquire information about a person and her/his family.

Genealogy can be found in all nations and historical periods – either in as a hobby, or as a profession. Genealogy has existed in one form or another since ancient times. The study of families and the tracing of their lineages and history has been a must for the rulers, aristocracy and wealthy people not only for preserving their family history, but also from preventing claims for their thrones, titles and wealth.

Along with the aforementioned in the last paragraph, it should be noticed that genealogy was important for the historiographers, who studied the life of historical figures from various fields of human knowledge. For them, this was a part of their



professional activities and in this way they were the first people who scientifically studied the pedigree, although they have not considered the genealogy as a separate field of science.

Along with the prosperity grow of the population in various countries, genealogy started to gain popularity among the middle class as a hobby – for example, in order to consolidate the family relations. In the end of the XIX century, the first genealogy societies were created and the genealogy gradually started to be considered as a separate field of science. Also, the term genealogy gained more and more popularity. In the XX century the interest in genealogy was growing more and more, because of the easy access to information and its popularization through various TV series. The increased interest naturally led to the need of inventing tools for automation of the activities, connected with the genealogy. So, in the beginning of the 80s the first publicly available genealogy software products came into existence.

1.2 Genealogy Software Overview

The first genealogy software products were developed for the purposes of corporations/organizations or for personal use. Such software products date back from the end of the 60s to the beginning of the 70s of the XX century and were predecessors of the commercially available genealogy software products, which came into existence almost immediately after the personal computers became available for the general public – in the very beginning of the 80s of the XX century (as mentioned in the previous section). Some of the most popular software products in these early years of the genealogy software were *ROOTS/M* and *Family Roots*. These products, as well as the other genealogy products of this time, were coded in the BASIC programming language and were available on cassettes and later on diskettes [3, 4].

Only for 10 years – till the end of the 80s – the number of available genealogy software products significantly increased, along with the possibilities they were providing the users, while their prices were decreasing [5].

This tendency continued in the 90s [4] and continues nowadays [3, 6, 7, 8, 9], which is proved by the currently available genealogy software products [6, 7, 8, 9]. Namely some of these products are the main subject of the current publication.

2 Review of Popular Genealogy Software Products

Taking into consideration the large number of existing genealogy software products, it is clear that in this publication only some of these software products could be reviewed. This means that in first place some selection of genealogy software products should be made.

In order to make a correct choice, four leading and independent genealogy web sites [6, 7, 8, 9], providing genealogy software reviews will be used. On the base of the research results, presented on these web sites, four genealogy software products are selected for the current review. The selected genealogy software products were chosen, because they are the most popular/quality products, according to the criteria used by this four leading web sites. The selected genealogy software products are the

following (in alphabetical order): *Family Tree Maker*, *Legacy*, *RootsMagic* and *The Master Genealogist*. Every of these software products will be reviewed in a separate section. All reviews in section 2 are compiled from the already mentioned web sites, namely [6, 7, 8, 9]. Before these sections, the review plan is presented in the next.

2.1 Review Plan

The review plan is compiled on the base of the aforementioned four sites [6, 7, 8, 9] and consists of the following items:

Feature set. The item *Feature set* presents the capabilities and specific characteristics of the reviewed software product.

Ease of Use. The item *Ease of use* of the software product describes to what extent it is user friendly, i.e. to what extent it is easy to use both for amateur and professionals.

Reporting and Publishing. The item *Reporting and Publishing* presents the output capabilities of the reviewed software product, i.e. whether the software product supports various output formats (generating web pages, for example).

Help/Documentation. The item *Help/Documentation* describes the help system and the documentation of the software product, i.e. whether the help needed can be easily found and useful.

Summary. The item *Summary* contains conclusion words and lists its pros and cons.

2.2 Family Tree Maker

Feature Set. The latest version of Family Tree Maker – Family Tree Maker 2010 – is a software product created to help to various kinds of users in exploring their family history. The workspace navigation is simple and provides the user with a various tools for exploring her/his family history and creating a family tree. The main part of the work could be completed on the People Workspace. From this workspace the user can do three main activities: create and modify an individual's profile, see a list of all the entered names and see the family tree diagram. When there are common facts in the family tree, the Family Tree Maker will link the shared details with all of the persons who are related to it (for example, viewing the spouses in one marriage). To assist the user in the input of correct information, the product provides an automated system that will alert her/him for data duplication or incorrect dates.

Via creating a paid registration to Ancestry.com, the user could do an online family research. This online database integration is a unique feature of this software product and is not presented in other genealogy software products on the market. This feature brings in many possibilities to find sources around the world. It should be noted, that

there is no assurance that the data will match the relative accurately. To avoid this problem from happening, a ranking scale from one to five has been used to give the source trustworthiness. A green leaf is a symbol that will show on the name of persons that have more data available through the genealogical sources online. When the user discovers facts of her/his relatives in the web, she/he can simply click and drag the facts found to the workspace and they will be automatically linked as a source.

As the family tree of the user develops into a large one, she/he may wish to make use of the Places Workspace. This workplace provides a possibility to review all of the locations that are linked to the relatives. Clicking on the location displays all the information connected with that location (for example, births, deaths or marriages). The map is provided by Microsoft Bing. It may be presented in three different ways: satellite photograph, road map, or a mixture of the two.

Except for the aforementioned basic features, the software product provides a number of innovative features that, though not crucial for the process of creating and maintaining the family tree, could be quite useful. These innovative features are presented in the next three paragraphs.

Important World Events. Important world events may be added to person's timelines that will help one to get an orientation of what may have formed their lives. In order this process to be made easy, natural disasters, wars, and other historical events are structured by year along with the person's age at the time when the event occurred.

Merging Data. Merging data is made easy via the Web clipper, which is used to retrieve online data concerning your relatives.

3D Option. In order to present a view of the locations near to their real look, there is a 3D option that shows a 3D view of the buildings and other objects.

Ease of Use. Family Tree Maker has a layout and design that will help anyone to make their genealogy exploration regardless of how much practice they may or may not have. Family Tree Maker makes the process of importing a family tree set up with another program fairly easy. The People workspace is a unique feature that is not available in other genealogy software products. This workspace allows having a mixed view of the family and pedigree. Even beginners will find it very easy to use Family Tree Maker.

Reporting and Publishing. There is a wide assortment of reports and charts to choose from when printing. Some of them are the following: pedigree, descendant, relationship.

Reports could be created for people, relationships, places, media and sources. Books can also be published. The latter requires visiting Ancestry Press, the online publishing site.

The charts can be personalized – for example, there are various backgrounds to choose from, including a user’s photo. The images of the persons may be placed next to their name, thus creating a meaningful display of the user’s ancestors.

Creating web pages is a standard feature in most of the genealogy software products and, as could be expected, Family Tree Maker provides it, too, giving a various possibilities for customizing the process of web pages creation.

Help/Documentation. The Family Tree Maker Web Site – FamilyTreeMaker.com – gives access to numerous tutorials that can be used as walkthrough for studying all the features of the software product. Along with it, the user receives a full length DVD. Creating and exploring the family history is assisted by built-in prompts. In the online knowledge base the user can search through useful clues of any subject and if they do not help her/him, customer service contact information is provided.

Summary. Family Tree Maker 2010 is an excellent genealogy software product with a database that will be helpful for any genealogist. This genealogy software product is one of the bestselling products of this type. No matter if the user is a beginner or has a lot of experience, she/he will find this software product as one of the best available on the market.

The *main pros* are the integrated online database, people workspace, new higher resolution maps with GPS coordinates, source citations template and rating system.

The *main con* is that the GEDCOM¹ interface needs to be improved.

2.3 Legacy

Feature Set. The latest version of Legacy – Legacy 7 – is organized around four family tree views. The family tree can be imported from a GEDCOM file or be built from scratch. The individuals can be added on the Family View or on the Pedigree View and the information about the particular person is accessed via a click over the individual. The icons could be used for adding additional information (i.e. spouses, siblings and children). Life events could be browsed from a drop down menu. The items in this menu could be directly added, or edited before addition. It is also possible new events to be created and added to a person. All new events are kept in the Master Event List. Both persons and information can be marked as private. The items marked as private will not be included in the created reports and charts. Tracing predecessor lines is simplified by using a color-coding. The various available tools in the homonymous menu show the flexibility and customization options of this product.

Legacy allows rating each source of the fact on a five levels scale – from zero to four („have not decided yet” ... „convincing evidence”). If the users do not need rating, she/he can only check the checkbox to notice that the source validity has been confirmed.

After adding the existing information for the family, a research should be made. The research in Legacy is done with the Research Guidance tool, which includes

¹ GEDCOM, an acronym of GENEalogical Data COMMunication, is a de facto specification for exchanging genealogical data between different genealogy software products. [10]

several components. The research starts with a Timeline Review and ends with a To-Do List with in depth source tips during all steps. Source correctness is important, thus Legacy provides tools to aid verify the information. Legacy could produce a Potential Problems Report noticing discrepancies (for example death after 120 years old or potential mistakes in a person's title in the name field). The user chooses which problems to verify and which persons to include in the report. Legacy has also a feature for searching missing information, which permits the user to search her/his family history and find the missing information (for example birth/death date or birth/death place).

Legacy also incorporates tools for producing various types of charts, displaying the user's family tree, including DNA charts. Legacy uses Microsoft Virtual Earth to enable the automatic location of important places and tracing the predecessors' resettlements/voyages.

Except for the aforementioned basic features, the software product provides a number of innovative features that, while not crucial for the process of creating and maintaining the family tree, could be quite useful. These innovative features are presented in the next three paragraphs.

Address Book. Legacy allows creating an address book and using it for printing the address labels and nametags along with photos and three-generation pedigree charts. The home page of Legacy automatically retrieves birthdays and anniversaries from the family tree information and includes it in the address book.

Intellishare Option. The Intellishare option manages group updates to the family file when it is being maintained by several people. In this case a Keeper of the Family File should be assigned in order to make the coordination between participants. The Keeper uses Legacy's Intellishare option in order to merge and reunite records at their discretion.

Windows Mobile Devices Synchronization. Legacy Family Tree 7 supports synchronization with Pocket Genealogist, the leader in genealogy software for Windows Mobile devices.

Ease of Use. No matter whether the user is into genealogy or not, she/he should be able to easily master the workflow process with this product. The user has complete control over the software – she/he could decide which tools to use and how to use them at every single step of the workflow process.

Reporting and Publishing. Legacy provides a large number of reports including Descendant Narrative, LDS Ordinances and TempleReady Instructions, and Family Picture Tree in Legacy's Report pane. Blank forms are also available. All the reports can be saved as a PDF file in order to be easily shared via the internet or directly printed on the printer, connected with the user's computer.

Legacy 7 offers new charting abilities with a user-friendly interface and additional types of charts. The chart formats include ancestor, descendant, fan, hourglass, bow

tie, DNA charts, and more. Each format is illustrated with colorful images. The appearance of the chart can be customized with themes, colors, text and format, lines, borders, and backgrounds. The charts can be printed, emailed, exported to a file or be made as a wall chart from Legacy.

In the Publishing Center, books with selected reports can be created. Legacy will automatically create the table of contents, names index, preface page and other. There is a full control over design and layout. Books and other reports can be sent to the printer or published in several formats: Text, HTML, Rich Text and PDF.

Creating web pages is a standard feature in most of the genealogy software products and, as could be expected, Legacy provides it, too, giving a various possibilities for customizing the process of web pages creation.

Help/Documentation. There is an integrated video tutorial in the Research Guidance, an informative guide included in Legacy's help. Text help is accessible on each screen. The help is searchable and provides full instruction on every task. Online, the user can find frequently asked questions, technical support, tips, and user forums. Two volumes of video training CDs and a 336 page manual could be bought online (from the product's site).

Summary. Legacy provides from simple to complex tools. After acquiring its full capabilities, the user could do the work of a professional genealogist. Legacy also provides one of the most comprehensive printing tools available, thus offering the ability a variety of charts and displays to be produced.

The *main pros* are the learning easiness, availability of a basic free version, possibility to create elaborate books and web pages and guided research feature.

The *main con* is the lack of detail in pedigree view.

2.4 RootsMagic

Feature Set. The latest version of RootsMagic – RootsMagic 4 – allows adding a limitless number of facts for every family member. There are numerous predefined events, including Bar Mitzvah, Blessing and Birth; Education, Election, Excommunication and many others. If the user can not find the fact she/he needs, a new event could be added. The addition includes setting a name for the event, a usage style in a sentence in the created reports, and setting which reports it should be included in. Checking the work done is achieved by running a Problem Search, which could be found under the Tools menu. This tool will catch relationship problems, but do not detect typographical errors.

Sourcing all the information is crucial in every careful research, especially when family history is considered. RootsMagic has a feature to notice whether the information is accurate, but there is no grading systems. RootsMagic also allows adding unrestricted number of source citations to every fact and one source could be easily applied to several facts.

After creating address book, it could be send via an email with single click. Also, some simple photo corrections could be done within RootsMagic and then be available for usage in charts and reports.

RootsMagic does not include a guided research tool, and if the user wants to use it, she/he should bought GenSmarts, a guided research program that automatically integrates with RootsMagic. This program offers sources for supplementary facts about selected family members. Images show whether the source is free or paid. Once GenSmarts is downloaded and installed, it could be found under the Tools menu.

Unlike the already reviewed software products, RootsMagic 4 does not provide innovative features.

Ease of Use. Information entry and importing are effortless and the workspace is limited to five views: pedigree, family, descendants, people and WebSearch. This basic approach, however, is misleadingly simple. RootsMagic Explorer, an essential tool in RootsMagic, needs an understanding of logic which could be difficult. This feature is for searching within the family tree, so it is difficult trees with many documented persons to be searched.

Reporting and Publishing. RootsMagic provides a variety of reporting and publishing capabilities. They include pedigree charts, family group sheets, narrative reports, individual summaries, blank forms, and more. Decorative charts like fan charts and large formats can be printed via PrintMyChart.com, RootsMagic's online partner. RootsMagic allows compiling reports into a book via the Publisher function.

Creating web pages is a standard feature in most of the genealogy software products and, as could be expected, RootsMagic provides it, too, giving a various possibilities for customizing the process of web pages creation.

Help/Documentation. RootsMagic has sufficient Help resources. The Help could be browsed by topics or could be searched. A list with frequently asked questions is also available. The users could also get support through an online form.

Summary. RootsMagic is a relatively good program, especially for beginners. Though, no innovative features could be found, so the program is not the best choice for professional research. Also, the complex searches within the family trees are not simple, which is another con for the professional genealogists.

The *main pros* are the learning easiness, availability of a basic free version, possibility to create shareable CDs and web pages.

The *main cons* are the absence of innovative feature, complex search function, absence of typographical errors detections.

2.5 The Master Genealogist

Feature Set. The Master Genealogist is created to manage large volumes of research information, photos, and sources. Tags can be attached to every family individual. In other software these tags are called facts or events. In the latest version – The Master

Genealogist 7 – the user could find over one hundred Source Types to choose and edit.

After the available information for the user's family tree has been entered, the research work could begin. Under Web, Other Sites, The Master Genealogist links to nine research sites. The user could add other sites to this list. The Master Genealogist does not provide an integrated database as some of the other genealogy programs.

Searching the tree is made by using a variety of filters. Picklist is the method for quick search method – for finding a particular individual. With Project Explorer more complex searches could be carried out – for groups of people.

The Master Genealogist includes a large number of historical events added by users, thus allowing them to see their predecessors in the context of the world events. There are predefined timelines and user can edit any of them and add her/his own.

Users could use a Statistical Report to compare any group within the scope of the user's family to another group, males versus females or one family to another. Some of the created statistics are average age at first marriage, first and last child, and death.

The Master Genealogist could import data from practically all the other popular genealogy software – a feature that can not be found in other products.

It should be noted that the number of features provided by The Master Genealogist is immense, because it is created mainly for professional genealogists.

Ease of Use. The very steep learning curve is the main con of The Master Genealogist. This is a complex program, which relies on relatively outdated design. Some simple changes would make working with The Master Genealogist much easier without reducing its power. Examples for such changes are using informal labels and language along with the formal one and offering preview narratives in source code edits along with the computer coding.

Reporting and Publishing. The Master Genealogist has the most extensive report menu available. Genealogy experts could find sophisticated reports like Ahnentafel charts, Kinship Reports, Fan Charts. As expected, all of the standard reports/charts are provided. A Book Manager is available – it generates compilation of specific reports.

Creating web pages is a standard feature in most of the genealogy software products and, as could be expected, The Master Genealogist provides it, too, giving a various possibilities for customizing the process of web pages creation.

Help/Documentation. There is a detailed help section which could be browsed by topics or words. Tips and Hints, an Online Guided Tour, frequently asked questions, are available in the Help section. The users could get support via email or telephone.

Summary. The Master Genealogist is created mainly for the most serious genealogists who research family history with discipline and have proficiency in genealogy (genealogy scientists, for example). Thus, The Master Genealogist 7 is the best choice if the user is a professional and/or would like to record every possible detail with the most flexibility. Though this genealogy software product is a powerful tool, all reviewers agree that it is not the best choice for beginners or average

researchers. The Master Genealogist has such a steep learning curve that the user should buy books and videos in order to use its full potential.

The *main pros* are the capability of recording every possible detail with the most flexibility, sophisticated publishing formats, capability of recording and noting conflicting data, capability to import data from virtually all the other popular genealogy products, existence of a Mac-version, capability to create web pages.

The *main con* is the very steep learning curve.

3 Conclusion

The first three software products are the most popular among the non-professionals and the tendency is to include more and more features in order to become more popular amongst the professionals, too. Notwithstanding the increasing number of features, these products provide user-friendly interface, even for beginners.

The last software product is the one with most features from all the other genealogy software products on the market. It is the best choice for the professional genealogists, who would like to record every possible detail and to import/export from/to various formats, including scientific. Working with so extensive number of features, however, requires knowledge of genealogy terminology, as well as specific IT skills, which makes this product not appropriate for beginners/casual users.

References

1. Encyclopedia Britannica article on genealogy. Last visited March 9, 2010, from Encyclopedia Britannica Web site: <http://search.eb.com/eb/article-9108628>
2. Wikipedia article on genealogy. Last visited March 9, 2010, from Wikipedia Web site: <http://en.wikipedia.org/wiki/Genealogy>
3. Eastman, D.: Early Genealogy Programs: A Look Back at the Humble Beginnings of Genealogy Software (2002). Last visited March 9, 2010, from About.com Web site: <http://genealogy.about.com/library/authors/uceastman1a.htm>
4. Clay, B.: Computers and Genealogy: What Was, Is, and May Be. In Genealogical Computing Vol. 19, No. 2 (1999). Last visited March 9, 2010, from Ancestry Web site: <http://www.ancestry.com/learn/library/article.aspx?article=1644>
5. Pence, R.A.: Summary of Genealogy Software (1993). Last visited March 9, 2010, from Ancestry Web site: <http://ftp.rootsweb.ancestry.com/pub/roots-1/faq/faq.pc>
6. Genealogy Software Reviews. Last visited March 9, 2010, from the TopTenReviews.com website: <http://genealogy-software-review.toptenreviews.com/index.html>
7. Genealogy Software Reviews. Last visited March 9, 2010, from the consumersearch.com website: <http://www.consumersearch.com/genealogy-software>
8. Genealogy Software Reviews. Last visited March 9, 2010, from the genealogyreviews.co.uk web site: <http://www.genealogyreviews.co.uk/genealogysoftware.htm>
9. Genealogy Software Reviews. Last visited March 9, 2010, from the gensoftreviews.com web site: <http://www.gensoftreviews.com/index.php>
10. Wikipedia article on GEDCOM. Last visited March 9, 2010, from the Wikipedia web site: <http://en.wikipedia.org/wiki/GEDCOM>

Author Index

- Atanassov Krassimir, *Bulgarian Academy of Sciences* 191
- Blazevska-Tabakovska Natasa, *Bitola University „St.Kliment Ohridski“, Macedonia* 225, 234
- Bontchev Boyan, *University of Sofia “St. Kl. Ochridski”* 39
- Cherian Sherimon, *Higher College of Technology of Muscat, Oman* 9
- Covacheva Zlatinka, *Higher College of Technology of Muscat, Oman* 9
- Dimitrov Vladimir, *University of Sofia “St. Kl. Ochridski”* 18, 125,164
- Dzhosan Oxana, *Lomonosov Moscow State University, Russia* 115
- Fidanova Stefka, *Bulgarian Academy of Sciences* 191
- Georgiev Vasil, *University of Sofia “St. Kl. Ochridski”* 133
- Girija N., *Higher College of Technology of Muscat, Oman* 30
- Goranova Radoslava, *University of Sofia “St. Kl. Ochridski”* 245
- Hristov Hristo, *University of Sofia “St. Kl. Ochridski”* 258
- Hristov Nikola, *Technologica Co, Bulgaria* 93
- Iliev Oliver, *Bitola University „St.Kliment Ohridski“, Macedonia* 225
- Ismaili Florije, *South-East European University, Macedonia* 105
- Kalin Georgiev, *University of Sofia “St. Kl. Ochridski”* 145
- Kaloyanova Kalinka, *University of Sofia “St. Kl. Ochridski”* 93
- Kirov Nikolay, *New Bulgarian University* 218
- Kolevska Silvana, *Bitola University „St.Kliment Ohridski“, Macedonia* 64
- Korzhan Anton, *NICEVT Co, Russia* 115
- Krishnan Reshmy, *Higher College of Technology of Muscat, Oman* 9
- Krustev Evgeny, *University of Sofia “St. Kl. Ochridski”* 72
- Manevska Violeta, *Bitola University „St.Kliment Ohridski“, Macedonia* 64, 234
- Marinov Pencho, *Bulgarian Academy of Sciences* 191
- Miloslav Sredkov, *University of Sofia “St. Kl. Ochridski”* 145
- Naydenova Ina, *Technologica Co, Bulgaria* 51

- Neykov Svetoslav, *University of Sofia "St. Kl. Ohridski"* 254
- Nisheva-Pavlova Maria M., *University of Sofia "St. Kl. Ohridski"* 200, 218
- Pavlov Pavel, *University of Sofia "St. Kl. Ohridski"* 209
- Peycheva Lozanka, *Bulgarian Academy of Sciences* 218
- Raufi Bujar, *South-East European University, Macedonia* 105
- Rendevski Nikola, *Bitola University „St.Kliment Ohridski“, Macedonia* 225
- Savoska Snezana, *Bitola University „St.Kliment Ohridski“, Macedonia* 64
- Serafimova Ana, *University of Sofia "St. Kl. Ohridski"* 263
- Serafimova Darina, *University of Sofia "St. Kl. Ohridski"* 273
- Todorova Magdalina, *University of Sofia "St. Kl. Ohridski"* 153
- Trifon Trifonov, *University of Sofia "St. Kl. Ohridski"* 145
- Vasileva Svetlana, *Shumen University „Bishop Konstantin Preslavski“, Bulgaria* 79
- Vinu P. V., *Higher College of Technology of Muscat, Oman* 30
- Vlahu-Gjorgievska Elena, *Bitola University „St.Kliment Ohridski“, Macedonia* 225
- Zenuni Xhemal, *South-East European University, Macedonia* 105
- Zhelev Radko, *Bulgarian Academy of Sciences* 133

