Pre-proceedings of the
First International Conference on

Interoperability of Enterprise Software and Applications

INTEROP-ESA'2005

Geneva, Switzerland, February 23 - 25, 2005

Doctoral Symposium

Preface

The interoperability in enterprise applications can be defined as the ability of a system or a product to work with other systems or products without special effort from the customer or user. The possibility to interact and exchange information with internal and external collaborators is a key issue in the enterprise sector. It is fundamental in order to produce goods and services quickly, at lower cost, while maintaining higher levels of quality and customisation. Interoperability is considered achieved if the interaction can, at least, take place at the three levels: data, applications and business enterprise through the architecture of the enterprise model and taking into account the semantics. It is not only a problem of software and IT technologies. It implies support of communication and transactions between different organisations that must be based on shared business references.

The INTEROP-ESA conference as well as the Doctoral Symposium focus on interoperability related research areas ranging like Enterprise Modeling to define interoperability requirements, Architecture and Platforms to provide implementation frameworks and Ontologies to define interoperability semantics in the enterprise.

The Doctoral Symposium is an opportunity for students involved in the preparation of a PhD in any area of Interoperability of Software and Applications to interactively discuss their research issues and ideas with senior researchers, and receive constructive feedback from members of the research community and other Doctoral Symposium students.

Subjects covered this year in the Doctoral Symposium relate to the following areas:

- Transactional Workflows and Workflows Modelling
- Ontology and Semantics
- Methods and Modelling techniques
- Interoperability of Software
- Security, Time and Knowledge Management
- Issues related to collaborative Entreprises

I wish to all authors an interesting and productive Doctoral Symposium, with space to expose their ideas, and time to create contact and initiate fruitful collaborations.

February 2005
Giovanna Di Marzo Serugendo
Doctoral Symposium Chair

Table of Contents

PART IV: Enterprise/Process Modelling

PART V: Interoperability of Software

PART VI: Management

PART VII: Collaborative Enterprises

PART VIII: Modelling

Customizable Isolation in Transactional Workflow

Adnene Guabtni[1], François Charoy[2], and Claude Godart[3]

[1] guabtni@loria.fr
[2] charoy@loria.fr
[3] godart@loria.fr
LORIA - INRIA - CNRS (UMR 7503) BP 239, 54506 Vandouvre-lès-Nancy Cedex, France

Abstract. In Workflow Management Systems (WFMSs) safety of execution is a main need of more and more business processes and transactional workflows are real needs inside enterprizes. In previous works, transactional models consider mainly atomicity as the main issue regarding long term transactions. It rarely consider the fact that many processes may run concurrently and thus access and update the same data. Usually, the main isolation item is the data on which we apply locking approaches and this attitude don't worry about process dimension. In this work we study more precisely what are the real isolation needs in workflow environment. To realize these needs, we define "Isolation Spheres" inspired from "spheres of control" proposed by C. T. Davies to make a separation of concerns between workflow design and transactional properties specification.

1 Introduction

Defining the transactional requirements of business processes is still an issue in today workflow models and systems. This is even more critical when the complexity of the process increases. It is the case for instance with cooperative process or with distributed and composed e-services. Today's models consider the relationship between transactional properties and processes as something very monolithic. A process is considered as a long term atomic transaction and an activity is considered as a short term transaction. In the workflow terminology, that means that a process is controlled by some kind of advanced transaction model that ensure either that the process terminates or that it can be compensated. The other assumption is that activities can be implemented as short term database transactions. This has an impact on the way processes and activities are defined and it requires that business process designer have some in-depth knowledge of transactional requirements. Moreover these models consider mainly atomicity as the main issue regarding long term transactions. It rarely considers the fact that many processes may run concurrently and thus access and update the same data. Some work has been done on this topic in a recent past (contracts/coo) but it has never been generalized to process.

In this paper we try to consider processes as the concurrent execution of sets of activities that may have different requirements regarding isolation. Usually, isolation in workflow systems is performed by the database system of WFMS. Databases use ANSI SQL isolation levels to define isolation requirements of a transaction on some data items. The problem is that workflow isolation requirements cannot always be satisfied by a database system. Contrary to database transactions, workflow transactions are defined and organized throw a process. At design time, we know exactly what are possible concurrent transactions and we want to make it possible to allow a transaction to adopt different isolation levels depending on concurrent transactions. This need appears when an activity requires an isolation level to access some data and many activities become unable to access or modify this data even if some of them don't really affect the transaction's correctness or the consistency of data. The way we choose to tackle this problem consists in separating concern between the process definition and its transactional requirements. We consider that a process must be defined independently of the transactional properties that we need to ensure. The process definition depends on the actual user activities and should reflect the actual company organization. Transactions reflects technical and consistency requirements and should not impact on this definition. To perform that, we inspire ourselves from the sphere of control approach proposed by C. T. Davies in [6] in 1978. This approach has been reused in 2001 to produce atomicity spheres in [4] to perform customizable atomicity specification in transactional workflow. We use the same approach to define isolation spheres to allow customizable isolation specification.

In the following sections of the paper, we study isolation needs in database world already applied to workflow processes. Next we try to specify transactional workflow isolation requirements. Finally we develop our approach based on isolation spheres to allow customizable isolation in transactional workflow.

2 Isolation needs in transactional workflow

Isolation is an important and difficult problem as it requires to consider access to data during process and activities execution. It requires to study data manipulation by process, activities and/or sets of activities. It requires also to take into account the fact that long term process execution cannot require locking of whole set of data for all its duration. The requirements regarding these data can be far more complex. Isolation levels in flat transactions has been recognized in ANSI SQL specification [1] where the user can choose between 4 different isolation levels : (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE) to prevent phenomena like dirty read, fuzzy read or phantom problems as described in the table below. Dirty read problem occurs when a transaction reads an uncommitted data rollbacked later. Non repeatable or fuzzy read problem occurs when a transaction reads a data two time and retrieve two different values. Phantom problem occurs when a transaction reads a set of data satisfying some search condition and then repeats its read with the same search condition, it gets a different set of data items.

Isolation levels	Dirty read	Fuzzy read	phantom
READ UNCOMMITTED	Possible	Possible	Possible
READ COMMITTED	Not possible	Possible	Possible
REPEATABLE READ	Not possible	Not possible	Possible
SERIALIZABLE	Not possible	Not possible	Not possible

Table 1. SQL isolation levels defined in terms of the three phenomena

WFMS and Databases don't refers to the same requirements. Workflow processes are based on a controlled flow of tasks but this control is not sufficient to ensure correct execution and don't prevent from lack of consistency. This is due to workflow data visibility that is a paramount way to distribute access to data in a workflow process but also a real source of concurrency access. In the next section, we expose isolation needs in workflow processes and what is important to do in the case of activities groups.

2.1 Isolation needs in WFMS

Data accessed during a process execution are heterogeneous. They consists in documents, folders, cases data, local data, database system data and/or data obtain from external sources. Access control on these data may be very different and may have different kind of impact on the level of isolation that can be obtain. Moreover access to these data can be controlled by automatic activities or by users themselves. The level of control differs also in these two cases. Execution of automatic programs can be anticipated. User action cannot. We need to take all these parameters in account to study isolation requirements in workflow processes.

Based on previous conclusions, we need to introduce new elements in the isolation levels use performed by the transactional workflow designer. These elements are the cohesion and the coherence on a group of activities. In the following, we describe these new workflow isolation behaviors

One of the needs of transactional workflow is the control of the **cohesion** of data used in a group of activities (collaborative work, distributed or composed E-services,). The solution used nowadays to ensure this cohesion of data for groups of activities is to create only one transaction imbricating all the others. Admittedly this approach makes it possible to ensure such a cohesion but has a major impact on the competition of access since it calls upon bolts in writing.

A second need is that of the **coherence** of the data. Indeed, the fact of allowing external activities with a group of reading certain data written by activities of that Ci can cause certain inconsistencies outside the group. It is for that that a control of the visibility of the data written by the activities of a group must be assured.

Related works were made in [7] to support partial isolation in flat transactions but it was made without a real separation of concerns. In the reality, relativity and extension of isolation are merged to express customizable requirements. These requirements are usually influenced by the requirements of each activity and the pertinence of the isolation is more and more crucial depending on the type of used data and its visibility in the workflow. In the next section, we introduce a new approach based on "isolation spheres" to take into account workflow isolation needs expressed in this paper.

3 Our approach : Isolation Spheres

In the last few years, some works has been inspired from the sphere of control proposed by Davies [6] to enhance expressivity of transactional properties, especially in [4] where the notion of atomicity sphere has been developed to allow more customizable atomicity in transactional workflow. A sphere of atomicity is a group of activities on which we apply the transactional property of atomicity. In our work, we inspire from this sphere of control approach and we define "spheres of isolation". A sphere of isolation will allow us to generalize isolation in the context of a workflow system. An isolation sphere allows the inside group of activities to be isolated from concurrent outside activities. The level of isolation is defined by the sphere. two kind of constraints are defined by the sphere : Coherence and Cohesion.

An isolation sphere controls the access to some data giving some privileges to a set of activities and some others to the rest of workflow activities depending on the execution evolution inside the set.

An isolation sphere represents a set of activities in concurrency working on some data. All or a part of this data represents the isolation data (data concerned by isolation on which necessary locks need to be applied). To perform cohesion and coherence of this data, we introduce some cohesion levels and some coherence levels :

Read Uncommitted : if an activity of the sphere reads a data, it can read only the last value written before the starting of the sphere or a value written by an activity of the sphere. Thus, the group of activities constituting the sphere starts from the same value.

Read Committed : if an activity of the sphere reads a data then it can read only the last **validated** value written before the starting of the sphere or a value written by an activity of the sphere.

Repeatable Read : As the Read Committed except that it is also concluded that the value of the data is not modified by an external activity as long as the sphere did not finish its execution yet. The end of the execution of a sphere occurs when all its activities finished their execution.

Srialisable: emulate an execution in series of the activities of the sphere with outside ones. This level makes it possible to ensure a serialisability between the sphere and the rest of the process but does not ensure it between the activities of the sphere.

To ensure Coherence, some coherence levels are defined in the following :

Atomic coherence : All the values of a data written by the activities of the sphere are visible outside of the sphere.

Selective coherence : Only the **validated** values written by the activities of the sphere are visible outside of the sphere.

Total coherence : Only the **last validated** value written by an activity of the sphere is visible outside of the sphere.

Imbrication of isolation spheres is possible throw imbrication of sets of activities. Imbrication is a powerful way to express more possibilities in isolation behavior. While isolation levels can be relative to a part of the process, we can generate isolation behavior dependent on execution progress due to isolation relativity over sphere imbrication.

The power of isolation spheres is the simplicity of interpretation : an isolation sphere is represented as a group of activities that need to be isolated from external activities and don't worry about internal concurrency (concurrency between activities of the group). Internal isolation, if needed, can be performed by imbricated isolation spheres. So the work performed by the workflow designer to specify isolation requirements is simplified.

This isolation sphere based transactional workflow take account of more possibilities to customize isolation and introduce more flexibility in isolation behavior. But isolation levels defined in the ANSI SQL specification have been criticized in [3] due to the lack of clarity in the interpretation of these isolation levels and the lack of response to some phenomena other then dirty read, fuzzy read and phantom. Since that, ANSI SQL specification has changed to be SQL 3 but without changes in isolation levels. Non SQL isolation levels have been proposed as cursor stability isolation or snapshot isolation. We need to study the impact of using these isolation approaches on the isolation sphere definition.

4 Conclusion and perspectives

In this paper, we have focussed on isolation in transactional workflow. Existing approaches use techniques of isolation adapted to databases and this practise is not really adapted to workflow context. A specific adaptation of isolation techniques to transactional workflow increases expressivity in term of isolation and allow process to get rid of long blocking due to database isolation methods. Our study of the problem revealed two main isolation functionalities to make part of the transactional workflow possibilities : Cohesion to make possible the activities of a group to start working from the same values of data and become unified along the sphere execution, and Coherence to make it possible to control the delivery of data values to external activities. Our approach to make these two functionalities possible is based on "Isolation Spheres" inspired from Sphere of control introduced by C. T. Davies.

This work need to be continued referring to many aspects : the relation between isolation spheres declaration and the control flow of the workflow process, the correctness of imbricating spheres and the flexibility criterion that we need to find to ensure that a transaction will be performed with less blocking then before. Also an implementation of "isolation sphere" functionalities need to be performed in a WFMS to validate the feasibility of this work.

References

1. Ansi x3.135-1992, american national standard for information systems - database language - sql. November 1992.
2. W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.
3. Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, and Patrick O'Neil. A critique of ansi sql isolation levels. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 1–10. ACM Press, 1995.
4. Wijnand Derks, Juliane Dehnert, Paul Grefen, and Willem Jonker. Customized atomicity specification for transactional workflow. In *Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'01)*, pages 140–147. IEEE Computer Society, 2001.
5. Adnene Guabtni and François Charoy. Multiple instantiation in a dynamic workflow environment. In Anne Persson and Janis Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Lavtia*, volume 3084 of *Lectures Notes in Computer Science*, pages 175–188. Springer, Jun 2004.
6. Charles T. Davies Jr. Data processing spheres of control. *IBM Systems Journal 17(2): 179-198*, 1978.
7. Randi Karlsen. Supporting partial isolation in flat transaction. In *Proceedings of the 10th International Conference on Database and Expert Systems Applications*, pages 698–711. Springer-Verlag, 1999.
8. Nick Russell, Arthur H. M. ter Hofstede, David Edmond, and W.M.P. van der Aalst. Workflow data patterns. Technical Report FIT-TR-2004-01, Queensland University of Technology, Brisbane, Australia, April 2004.

Business Process Intelligence : Discovering and Improving Transactional behavior of Composite Services From Logs

Walid Gaaloul

LORIA - INRIA - CNRS - UMR 7503
BP 239, F-54506 Vandœuvre-ls-Nancy Cedex, France
gaaloul@loria.fr

Abstract. One of the main features that makes Web services a promising technology for automating B2B interactions is the ability to dynamically combine a set of services into a new value added composite Web services (CS). However, a main problem that remains is how to ensure a correct and reliable execution. This paper presents a set of techniques to improve CS transactional behavior for a better failures handling. Basically, we propose a set of mining techniques to discover CS model and its transactional behavior from logs. Then, based on this mining step, we use a set of rules to improve its design. We refer to this set of techniques as the Business Process Intelligence (BPI) techniques suite.

1 Introduction

Nowadays, enterprises are able to outsource their internal business processes as services and make them accessible via the Web. Then, they can dynamically combine individual services to provide new value-added composite services (CSs). However, due to the inherent autonomy and heterogeneity of services, a fundamental problem that remains concerns the guarantee of correct and reliable execution.

In this paper, we present a different approach that starts from a CS executions log and uses a set of mining techniques to discover the CS model and the CS transactional behavior. Then, based on these mined information, we use a set of rules to improve the CS failures handling and recovery. Discovering the transactional behavior allows to detect gaps mentioned above to improve the application reliability and interoperability. Our approach starts from effective executions, while previous works use only specification properties (which remain ”assumptions”).

The remainder of this paper is organized as follows. Section 2 presents the notion of composite Web service from a transactional point view. In section 3, we introduce some distinctive concepts and needed prerequisites. After that, we detail our approach that mainly proceeds in two steps. The first one consists in discovering the CS patterns from an event-based log (see section 4). Then, based on this mining step, we use a set of rules to improve the CS transactional design (see section 5). Finally, we conclude our paper by summarizing the main contributions and presenting some future works.

2 Transactional composite Sevices

Transactional composite services (TCS) is a CS that emphasizes transactional behavior for failures handling and recovery. Within TCS, we distinguish between *the control flow* and *the transactional behavior*.

2.1 Control flow

A process definition is composed of services. Services are related together to form a control flow via transitions which can be guarded by a control flow operator. The control flow dimension is concerned with the partial ordering of services. The services that need to be executed are identified and the routing of cases along these services is determined. Conditional, sequential, parallel and iterative routing elements are typical structures specified in the control flow dimension. We use workflow patterns [1] to express and implement the control flow dimension requirements and functionalities.

2.2 Transactional Behavior

CS transactional behavior specifies mechanisms for failures handling. It defines *service transactional properties* and *transactional flow (interactions)*.

services transactional properties : Within (TCS), services emphasizes transactional properties for its characterization and correct usage. The main transactional properties that we are considering are *retriable*, *compensatable* and *pivot*[2]. An service a is said to be retriable (a^r) *iff* it is sure to complete. a is said to be compensatable (a^{cp}) *iff* its work can be semantically undone. Then, a is said to be pivot (a^p) *iff* its effect can not be compensated.

Transactional flow : A transactional flow defines a set of interactions to ensure failures handling. Transactional CSs take advantage of service transactional properties to specialize their transactional interactions.

3 Prerequisites

3.1 TCS Event log

Following a common requirement in the areas of business processes and services management, we also expect the composite services to be traceable, meaning that the system should in one way or another keep track of ongoing and past executions. As shown in the UML class diagram in figure 1, `TCSLog` is composed of a set of events streams (see Definition 1). Each events stream traces the execution of one case (instance). It consists of a set of events (`event`) which capture the services life cycle performed in a particular TCS instance. An `Event` is described by the service identifier that it concerns, the current service state (`aborted`, `failed`, `canceled`, `completed` and `compensated`) and the time when it occurs (`TimeStamp`).

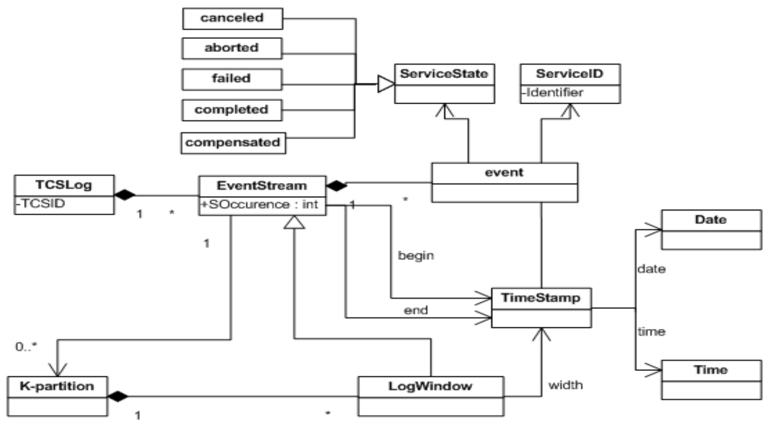


Fig. 1. Structure of a TCS Log

Definition 1 ***TCSLog***
*A TCS log is considered as a set of events streams. Each events stream represents the execution of one case. More formally, an events stream is defined as a quadruplet **stream**: (beginTime, endTime, sequenceLog, SOccurence) where:*

✓*(beginTime: time) and (endTime: time) are the moment of log beginning log end,*
✓*(sequenceLog: {**event**}): is an **ordered** events log tracing one TCS instance,*
✓*(SOccurence : int) is the service execution instance number.*

*So, **TCSLog**: (TCSID, $\{ServiceStream_i$: **stream**; $0 \leq i \leq$ number of TCS instantiations$\}$) is a TCS log where $ServiceStream_i$ is the events stream of the i^{th} TCS execution case.*

A `LogWindow` (see Definition 2) defines a set of `events` over an `EventStream`. Formally, the window beginning and ending represents an events stream interval characterizing the *width* of the `LogWindow`. A `K-partition` (see Definition 3) builds a set of partially overlapping `LogWindow`s partition over an `EventStream` where the width of `LogWindow`s is K. Each window is built by adding the next event log not included in the previous window.

Definition 2 ***log window***
*A log window defines a set of events over an events stream S : **stream** (bStream, eStream, sLog, TCSocc). Formally, we define a log window as a triplet **window**(wLog, bWin, eWin), where:*

✓*(bWin : time) and (eWin : time) are the moment of the window beginning and end (with bStream $\leq$ bWin and eWin $\leq$ eStream)*

✓*wLog $\subset$ sLog and $\forall$ e: **event** $\in$ S.sLog where bWin $\leq$ e.TimeStamp $\leq$ eWin $\Rightarrow$ e $\in$ wLog.*

Definition 3 ***K-partition***
*A **K-partition** builds a set of partially overlapping windows partition over an events stream.*
*K-partition : **TCSLog** $\rightarrow$ ({ **window**})**

*S : **stream**(sLog, TCSocc, bStream, eStream)* $\rightarrow \{w_i$ *: **window**; 1≤i≤n*$\}$*; where:*

✓ w_1*.bWin = bStream and* w_n*.eWin = eStream,*

✓$\forall w$: ***window*** $\in$ ***K-partition***, width*(w)= K,*

✓$\forall$ *0≤i<n;* w_{i+1}*.wLog -* $\{$*the last e:**event** in* w_{i+1}*.wLog*$\} \subset w_i$*.wLog and* w_{i+1}*.wLog* $\neq w_i$*.wLog.*

3.2 CS Set of Termination States

The state of a TCS instance composed of n services, at a specific time, can be presented by the tuple (x_1, x_2, ..., x_n), where x_i is the state of the service instance s_i at this time. A termination state of a TCS instance is the state in which this instance terminates. Let cs a TCS, we define $STS(cs)$ the set of all possible termination states of all cs instances.

4 Control Flow Mining

TCSs are case-based, many cases can be handled by following the same TCS process definition. Routing elements are used to describe sequential, conditional, parallel, and iterative routing thus specifying the appropriate route of a case. In this present work, we are interested in discovering ”elementary” routing TCS patterns: Sequence, AND-split, OR-split, XOR-split, AND-join, OR-join, and M-out-of-N Join patterns inspired from workflow patterns[1].

The main challenge which we cope with is the discovery of the sequential, conditional and concurrent behavior of these patterns. We meet these conditional and concurrent behavior after : (i) a ”fork” point where a single thread of control splits into multiple threads of control or before (ii) a ”join” point where multiple threads of control merge in a single thread of control. Thus, we have three types of *dependence relations* between services :

1. *Exclusive relation* is a activation dependency relation that captures the sequencing of services (*e.g.* Sequence pattern) where the enactment of a service depends only on the completion of one other.
2. *Conditional relation* is a activation dependency relation that captures selection of one service or more from a set of services potentially following after a ”fork” point (*e.g.* OR-split), or preceding before a ”join” point (*e.g.* OR-join), the execution of a given service.
3. *Concurrent relation* is not a activation dependency relation. In fact, it captures only concurrency or parallelism between services after a ”fork” point (e.g. AND-split pattern) and before a ”join” point (e.g. AND-join pattern).

Our control flow mining approach proceeds in two steps [3–5]: Step (i) the construction of statistical dependency table SDT, Step (ii) the mining of TCS patterns through a set of rules using the statistical specifications of these properties.

Construction of the statistical dependency table SDT : We use statistical calculus that extract control flow dependencies between services that are executed without "exceptions" (*i.e.* they reached successfully their **completed** state). There is no need to use others events streams relating to failure executions containing `failed` or `aborted` or `compensated` or `canceled` states. In fact, these cases concern only TCS transactional behavior (see section 2) which tailors the mechanisms for failures handling and recovery. For these reasons, we need to filter TCS log and take only events streams of instances executed "correctly". We denote by $TCSLog_{completed}$ this TCS log projection.

Thus, the minimal condition to discover TCS patterns is to have TCS logs containing at least the `completed` event states. This feature allows us to mine control flow from "poor" logs which contain only `completed` event state. Any information system using transactional systems such as ERP, CRM, or workflow management systems offer this information in some form [6].

For each service A, we extract From $TCSLog_{completed}$ the following information in the statistical dependency table (SDT): (i) The overall frequency of this service (denoted $\#A$) and (ii) The activation dependencies to previous services B_i (denoted $P(A/B_i)$). The size of SDT is N*N, where N is the number of TCS services. The (*m*,*n*) table entry (notation P(*m*/*n*)) is the frequency of the m^{th} service **immediately preceding** the n^{th} service. This SDT is a variation of the one from [7]. As it is computed, it presents some problems to express correctly services dependencies especially relating to concurrent or parallel behavior. In the following, we detail these issues and propose solutions to correct them.

Undetectable dependencies : If we assume that the events stream is exactly correct (i.e., contains no noise) and derives from a sequential (i.e no concurrent behavior) TCS, a zero entry in SDT represents an activation independence and a non-zero entry means an activation dependency relation (*i.e.* sequential or conditional relation). But, in case of concurrent behavior, as we can see in TCS patterns (like AND-split, AND-join, OR-join, etc) the events streams may contain interleaved events sequences from concurrent threads. As consequence, a service might not, for some concurrency reasons, depend on its immediate predecessor, but it might depend on another "indirectly" preceding service.

To unmask and correct this erroneous frequencies we calculate the frequency using a *concurrent window*, i.e. we consider not only the events occurring immediately backwards but also the preceding events covered by the *concurrent window*. Formally, a concurrent window defines a log slide over an events stream (see Definition 2). *The width* of the *concurrent window* is the maximal duration that a concurrent execution can take. It depends on the studied TCS and is estimated by the user. Based on that, we construct an events stream partition (see Definition 3). This partition is formed by a set of overlapping windows.

Based on this definition, we compute the new statistical service dependencies. we scan the set **K-partition window**s over **TCSlog**, **window** by **window**, and for each **window** we compute for the last service the frequencies of its preceded services and the corresponding table is updated in consequence. The statistic servive activation dependency will be found by dividing each row entry in the previous table by the frequency of its corresponding service.

Erroneous dependencies Some entries in SDT can indicate non-zero entries that do not correspond to dependencies. These entries are erroneous because there is no activation dependencies between these services as suggested. Formally, two services A and B are in concurrence *iff the P(A/B)* and *P(B/A)* entries in SDT are different to zero. Based on this definition we propose an algorithm to detect services parallelism and then mark the erroneous entries in SDT. Through this marking, we can eliminate the confusion caused by the concurrence behavior which produces these erroneous non-zero entries. This algorithm scans the initial **SDT** and marks the dependencies of concurrent services by changing theirs values to (-1).

4.1 TCS patterns mining

After the compute of the statistical dependency table, the last step is the identification of TCS patterns through a set of rules. In fact, each pattern has its own statistical features which abstract statistically its activation dependencies, and represent its unique identifier. These rules allow, if TCS log is completed, the discovery of the whole TCS patterns included in the mined TCS. To be complete, TCS log should cover all the possible cases (i.e. if a specific routing element can appear in the mined TCS model, the log should contain an example of this behavior in at least one case). But, our control flow mining rules are characterized by a ”local” TCS patterns discovery. Indeed, these rules are context-free, they proceed through a local log analyzing that allows us to recover partial results of mining TcS patterns. In fact, to discover a particular TCS pattern we need only events relating to pattern’s elements. Thus, even using only complete fractions of TCS log, we can discover correctly corresponding TCS patterns (which their events belong to these fractions). To be complete, these fraction should cover all the possible cases of these patterns.

We divided the TCSs patterns in three categories : sequence, fork and join patterns. In the following we present rules to discover the most interesting TCS patterns belonging to these three categories.

Sequence pattern : In this category we find only the sequence pattern, in witch the enactment of the service B depends only on the completion of service A (*c.f.* table 1).

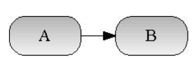
Rules	**workflow patterns**
$(\#B = \#A)$	Sequence pattern
$(P(B/A) = 1)$	

Table 1. Rules of sequence workflow pattern

Fork patterns : This category (*c.f.* table 2) has a ”fork” point where a single thread of control splits into multiple threads of control which can be, according to the used pattern, executed or not.

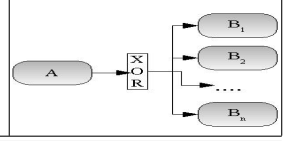
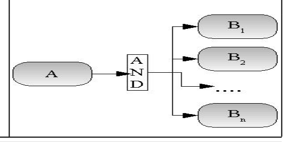
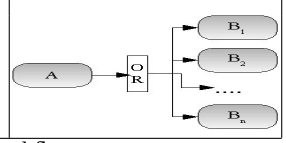
Rules	workflow patterns
$(\Sigma_{i=0}^{n}(\#B_i)$=#A)	XOR-split pattern
$(\forall 0 \leq i \leq n; P(B_i/A) = 1) \wedge$ $(\forall 0 \leq i, j \leq n; P(B_i/B_j) = 0)$	
$(\forall 0 \leq i \leq n; \#B_i$=#A)	AND-split pattern
$(\forall 0 \leq i \leq n; P(B_i/A) = 1) \wedge$ $(\forall 0 \leq i, j \leq n\ P(B_i/B_j) = -1)$	
$(\#A \leq \Sigma_{i=0}^{n}(\#B_i)) \wedge$ $(\forall 0 \leq i \leq n; \#B_i \leq \#A)$	OR-split pattern
$(\forall 0 \leq i \leq n; P(B_i/A) = 1) \wedge$ $(\exists 0 \leq i, j \leq n; P(B_i/B_j) = -1)$	

Table 2. Rules of fork workflow patterns

Join patterns : This category (*c.f.* table 3) has a "join" point where multiple threads of control merge in a single thread of control. The number of necessary branches for the activation of the service B after the "join" point depends on the used pattern.

5 Mining and Improving Transactional behavior

We define at this level a set of rules allowing to mine transactional behavior. These rules allow to tailor the services transactional properties according to the discovered control flow and set of termination states.
$\forall\ service\ s$:

1. $s.failed \notin STS(s) \Longrightarrow$ *s is retriable*
2. $s.failed \in STS(s) \Longrightarrow$ *s is not retriable*
3. $s.compensated \notin STS(s) \Longrightarrow$ *s is not compensatable*
4. $s.compensated \in STS(s) \Longrightarrow$ *s is compensatable* $\wedge$ ***have to be compensated when one of its compensation conditions occurs***; We extract from the discovered control flow and the set of terminated state the compensation condition of a denoted $cpsCond(a)$.

The first (respectively the second) rule says that if a never fails (respectively can fail) then a is (respectively is not) retriable. The third and forth rules allow to deduce when a service a is compensatable and when it will be compensated.

Mining transactional dependencies returns to mine corresponding preconditions of services external transitions. We can extract the potential compensation conditions for

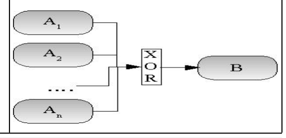
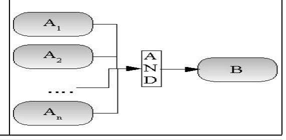
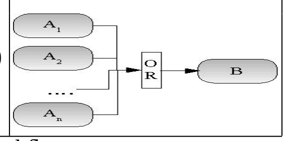
Rules	workflow patterns
($\Sigma_{i=0}^{n}$ ($\#A_i$)=#B)	XOR-join pattern
($\Sigma_{i=0}^{n}$ P(B/A_i)=1) $\wedge$ $(\forall 0 \leq i, j \leq n; P(A_i/A_j) = 0)$	
($\forall 0 \leq i \leq n$; $\#A_i$=#B)	AND-join pattern
$(\forall 0 \leq i \leq n; P(B/A_i) = 1) \wedge$ $(\forall 0 \leq i, j \leq n \; P(A_i/A_j) = -1)$	
($m * \#B \leq \Sigma_{i=0}^{n}$ ($\#A_i$)) $\wedge$ ($\forall 0 \leq i \leq n; \#A_i \leq \#B$)	M-out-of-N Join pattern
$(m \leq \Sigma_{i=0}^{n} P(B/A_i) \leq n)$ $\wedge$ $(\exists 0 \leq i, j \leq n; P(A_i/A_j) = -1)$	

Table 3. Rules of join workflow patterns

a given service s ($ptCpsCond_i(s)$) from the mined composite service skeleton and the the set of termination states (STS). The idea is : a potential compensation condition of s becomes a compensation condition if it is satisfied in a $ts \in STS$ such that the state of s in ts is compensated. We proceed similarly to deduce alternative ($ptAltCond_i(s)$) and cancelation conditions ($ptCnlCond_i(s)$) of each service.

Now, we can improve a TCS, regarding its initial design, through the following two phases : (i) omitting and correcting the wrong transactional mechanisms, (ii) and/or adding relevant transactional mechanisms for a better failure handling and recovery. By wrong transactional mechanisms we mean component transactional properties and transactional dependencies initially specified and which do not coincide with the reality. These wrong transactional mechanisms can be simply costly but also source of error. The following rules allow to generate the appropriate suggestions according to the discovered TCS transactional behavior.

$\forall$ component service, s, of TCS (we suppose that $\Diamond$F means F is eventually true):

1. $\forall \, ptCpsCond_i(s) \in ptCpsCond(s)$,
   $\Diamond(ptCpsCond_i(s)) \bigwedge ptCpsCond_i(s) \notin CpsCond(s) \Rightarrow$
   (a) s must be compensatable and
   (b) $CpsCond(s) = CpsCond(s) \bigoplus ptCpsCond_i(s)$.
2. $\forall \, ptCnlCond_i(s) \in ptCnlCond(s)$,
   $\Diamond(ptCnlCond_i(s)) \bigwedge ptCnlCond_i(s) \notin CnlCond(s) \Rightarrow CnlCond(s) = CnlCond(s)$
   $\bigoplus ptCnlCond_i(s)$.
3. $\forall \, ptAltCond_i(s) \in AltCond(s)$,
   $\Diamond(ptAltCond_i(s)) \bigwedge ptAltCond_i(s) \notin AltCond(s) \Rightarrow AltCond(s) = AltCond(s)$
   $\bigoplus ptAltCond_i(s)$.

$AltCond(s)$ defines the precondition to be enforced before the service s can be activated as an alternative of other service(s) (similarly for cancelation and abortion). The first rule postulates that for each potential compensation condition of s, $ptCpsCond_i(s)$; if this condition is eventually true and does not belong to the discovered compensation condition of s, then s must be compensatable and $ptCpsCond_i(s)$ becomes a compensation condition of s. That means $\forall$ s' $\in ptCpsCond_i(s)$ add a compensation dependency from s' to s according to $ptCpsCond_i(s)$. The second rule postulates that for each potential cancelation condition of s, $ptCnlCond_i(s)$; if this condition is eventually true and does not belong to the discovered cancelation condition of s, then $ptCnlCond_i(s)$ becomes a cancelation condition of s. That means $\forall s' \in ptCnlCond_i(s)$ add a cancelation dependency from s' to s according to $ptCnlCond_i(s)$. The third rule postulates that for each potential alternative condition of s, $ptAltCond_i(s)$; if this condition is eventually true and does not belong to the discovered alternative condition of s, then $ptAltCond_i(s)$ becomes an alternative condition of s. That means $\forall s' \in ptAltCond_i(s)$ add an alternative dependency from s' to s according to $ptAltCond_i(s)$.

6 Conclusion and Future work

In this paper we presented an original approach for ensuring reliable Web services compositions. Different from previous works, our approach starts from a TCS log. Generally, previous approaches develop, using their CS modelling formalisms, a set of techniques to analyze the composition model and check some properties [9–13]. Although powerful, the above formal approaches may fail, in some cases, to ensure CS reliable executions even if they formally validate the CS composition models. This is because properties specified in the studied composition models may not coincide with the reality (*i.e.* effective CSs executions). Note also that a number of research efforts in the area of workflow management have been directed for mining workflows models (a detailed survey of this research area is provided in [14]). This issue is closely to that we propose in terms of control flow discovery. But there are practically no approaches to transactional behavior workflow mining except in [8, 4].

Our approach uses a set of mining techniques to discover the TCS control flow and the TCS transactional behavior. Then, based on this mining step, we use a set of rules to improve the TCS design. The mining phase is itself divided into two steps. The first one consists in mining TCS patterns. Then from the discovered information, we use a set of rules to mine the TCS transactional behavior. Our approach starts from effective executions, while previous works use only specification properties (which remain assumptions). Our control flow mining approach is original regarding other proposed techniques. It is characterized by a ”local” discovery techniques that allows to recover partial results. In besides, it discovers more behavioral complex features with a better specification of ”fork” point and ”join” point.

However, the work described in this paper represents an initial investigation. In our future works, we hope to discover more complex patterns by using more metrics (*e.g.* entropy, periodicity, etc.) and by enriching the TCS log. We are also interested in the modeling and the discovery of more complex transactional characteristics of cooperative TCS.

References

1. W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.
2. A. Elmagarmid, Y. Leu, W. Litwin, and Marek Rusinkiewicz. A multidatabase transaction model for interbase. In *Proceedings of the sixteenth international conference on Very large databases*, pages 507–518. Morgan Kaufmann Publishers Inc., 1990.
3. W. Gaaloul, S. Alaoui, K. Bana, and C. Godart. Mining Workflow Patterns through Event-data Analysis. In *The IEEE/IPSJ International Symposium on Applications and the Internet (SAINT'05). Workshop 6 Teamware: supporting scalable virtual teams in multi-organizational settings*. IEEE Computer Society Press, 2005.
4. W. Gaaloul, S. Bhiri, and C. Godart. Discovering workflow transactional behaviour event-based log. In *12th International Conference on Cooperative Information Systems (CoopIS'04)*, LNCS, Larnaca, Cyprus, October 25-29, 2004. Springer-Verlag.
5. W. Gaaloul, S. Alaoui, H. Bakkali, K. Bana, and C. Godart. WorkflowMiner : An infrastructure for Mining Workflow Patterns. In *3mes Journes Nationales sur les Systmes Intelligents: Thories et Applications (SITA'04)*, pages 10–13, Rabat, Morocco, December 6-7, 2004.
6. W.M.P. van der Aalst and L. Maruster. Workflow mining: Discovering process models from event logs. In *QUT Technical report, FIT-TR-2003-03, Queensland University of Technology*, Brisbane, 2003.
7. Jonathan E. Cook and Alexander L. Wolf. Automating process discovery through event-data analysis. In *Proceedings of the 17th international conference on Software engineering*, pages 73–82. ACM Press, 1995.
8. W. Gaaloul, S. Bhiri, and C. Godart. Discovering workflow patterns from timed logs. In *EMISA 2004, Informationssysteme im E-Business und E-Government, Beiträge des Workshops der GI-Fachgruppe EMISA*, LNI, pages 84–94, Luxemburg, October 6-8, 2004. GI.
9. Tevfik Bultan, Xiang Fu, Richard Hull, and Jianwen Su. Conversation specification: a new approach to design and analysis of e-service composition. In *Proceedings of the twelfth international conference on World Wide Web*, pages 403–410. ACM Press, 2003.
10. Rachid Hamadi and Boualem Benatallah. A petri net-based model for web service composition. In *Proceedings of the Fourteenth Australasian database conference on Database technologies 2003*, pages 191–200. Australian Computer Society, Inc., 2003.
11. Benchaphon Limthanmaphon and Yanchun Zhang. Web service composition transaction management. In *Proceedings of the fifteenth conference on Australasian database*, pages 171–179. Australian Computer Society, Inc., 2004.
12. Paulo F. Pires, Mario R. F. Benevides, and Marta Mattoso. Building reliable web services compositions. In *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems*, pages 59–72. Springer-Verlag, 2003.
13. Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. Business process intelligence. *Comput. Ind.*, 53(3):321–343, 2004.
14. W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters. Workflow mining: a survey of issues and approaches. *Data Knowl. Eng.*, 47(2):237–267, 2003.

Topics abstract: Workflow Modelling and Enactment for Work Process Support on the Web

Rune Molden

Norwegian University of Science and Technology (NTNU),
Department of Computer and Information Science,
Sem Sælands vei 7-9, N-7491 Trondheim, Norway
rune.molden@idi.ntnu.no

Evolution of networked organisations [4] and project organised work [2] requires more advanced features for work process modelling than the current approaches offer. Utilisation of Semantic Web resources [6] in work process environments offers several advantages for adaptation and adoption of information and services. The models representing these processes should be the foundation of the real life executing processes and should be synchronised at any given time [2], even though needed resources for the processes are located outside the organisation. The resources may even be at an unavailable state at the time of usage. By utilising Semantic Web technologies such as RDF and OWL/OWL-S for enrichment of the information and service resources to use in work processes [7], a substitute service or information source can be utilised automatically by employing techniques such as ontology mapping and other criteria data for asymptotic exact mapping. Rating of services and information can also be of great value, e.g. to filter dubious information sources and services [1].

The Web Information Service Modelling (WISEMOD) project funded by the Research Council of Norway (NRF), in which this work is part of, has as one of its objectives to come up with a new and improved approach to workflow modelling, specifically tailored for the needs of Semantic Web application engineering [5]. This encompasses an approach for workflow modelling utilising Semantic Web applications which is to be used for both analysis and enactment of inter-organisational solutions [5]. Further, a methodology for employing the modelling approach is required and, finally, a modelling tool prototype is to be developed and evaluated [5].

The improvement of workflow modelling tailored for the needs of Semantic Web services, requests to be concerned with the different aspects in the nature of inter-organisational processes. Introductorily, it is essential to get a thorough overview and knowledge of the existing approaches for workflow modelling for utilising Web resources. Process modelling approaches like BPEL4WS, ebXML, BPML, BPMN, and WSCI will be thoroughly inspected and evaluated, and lessons should be learnt concerning strengths and weaknesses of these approaches. Further, the evaluation of the current process model standards should lead to a solid knowledge of the state of the art within this area of research. In addition, an investigation of the current Semantic Web technologies, standards and research within this area will be performed. These efforts will give a good basis for improvements and enlargement of the state of the art. More detailed, the approach will constitute the following aspects:

- Development of a new workflow modelling language and notation as an extension of existing languages and related technologies based on the state of the art approach and requirements elicitation/surveys in industry.
- Make a meta-model of the extended language and implement this through an existing meta-modelling, to experiment with the new modelling notation.
- Extend a current workflow enactment environment to meet the requirements of the modelling language and notation approach.
- The semiotic framework for quality of models and modelling languages [8] will be employed to analytically evaluate the extended workflow modelling language and model enactment environment.
- The modelling language, notation and the tool will be empirically evaluated using case studies in the industry utilising the new approach for workflow modelling.

The research work concerning the workflow modelling approach has currently not been started, but some initial state of the art work has been done. However, several contributions are anticipated in the project plan. Initially, the state of the art work aims to yield an evaluation framework to evaluate existing modelling approaches and tools based on current work on quality of models and modelling languages. Surveys of existing methodologies and tools for workflow modelling, and Web services with advanced properties will be performed by utilising the evaluation framework. The workflow modelling approach will yield an approach for workflow modelling in connection with Semantic Web applications utilised both for analysis and enactment of inter-organizational solutions, a methodology for using the modelling approach, e.g. in terms of guidelines, and, a modelling tool prototype.

References

1. Cardoso, J.: Quality of Service and Semantic Composition of Workflows. Ph.D. Dissertation. Department of Computer Science, University of Georgia, Athens, GA, USA (2002)
2. Jørgensen, H.D.: Interactive Process Models. Ph.D. thesis. Norwegian University of Science and Technology, Trondheim, Norway (2004)
3. Cardoso, J., Sheth, A.: Semantic e-Workflow Composition. Journal of Intelligent Information Systems. Volume 21, Issue 3 (2003) 191-225
4. ATHENA Public Web Site: What is ATHENA. http://www.athena-ip.org/ (2005)
5. The WISEMOD Project Web: http://www.idi.ntnu.no/~guttors/wisemod/ (2005)
6. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
7. Peer, J.: Bringing Together Semantic Web and Web Services. Lecture Notes in Computer Science, Vol. 2342. Springer-Verlag, Berlin Heidelberg (2002) 279-291
8. Krogstie, J., Sølvberg, A.: Information Systems Engineering: Conceptual Modeling in a Quality Perspective. Kompendiumforlaget, Trondheim, Norway (2003)

Ontology-Based Approach for Application Integration

Saïd Izza, Lucien Vincent, Patrick Burlat
Laboratoire G2I, Ecole des Mines de Saint-Etienne
158 Cours Fauriel, Saint-Etienne, France
izza@emse.fr, vincent@emse.fr, burlat@emse.fr

Abstract. Enterprise Application Integration (EAI) is still facing the crucial semantic integration problem. This latter is not correctly addressed by today's EAI solutions that focus mainly on the technical and syntactical integration. Addressing the semantic integration level will promote EAI by providing it more consistency and robustness. Some efforts are suggested to solve the semantic integration problem, but they are still not mature. This paper deals with semantic problem in EAI and will present an ontology-based approach in order to overcome some issues of the integration problem.

1. Introduction

In the last years, a new technology typically known as Enterprise Application Integration (EAI), have emerged as a field of Enterprise Integration [18]. In essence, EAI technologies provide tools to interconnect multiple and heterogeneous Enterprise Application Systems (EAS) such as ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), SCM (Supply Chain Management), and legacy systems. The most difficulty of the integration of these systems is that the latter were never designed to work together.

More recently and with the advent and the evolution of the Internet, Web Services (WS) have emerged and they provide a set of standards for EAI. Even if WSs are not fully mature, they seem to become the *linga franca* of EAI. This will notably make integration simpler and easier through using web protocols and standards.

Despite the whole range of available tools and widespread standards adoption, the main goal of EAI, which is the semantically correct integration of EASs, is not yet achieved. Indeed, EAI still provides technical and syntactical solutions but does not address correctly the semantic problem, which constitutes the real integration problem. Semantic integration becomes very important in order to overcame semantic heterogeneities within EAI, and which mainly concern both data and behavior of EASs. Although there is some related works, which concern semantic integration, but there has been no mature solution that deals correctly with integration problem.

In this paper, we will focus on the semantic problem in the context of EAI. Our approach is ontology-based and it can be seen as an extension of service-oriented architecture (SOA), it is called ODSOI (Ontology-Driven Service-Oriented Integration). The rest of this paper is organized as follows. We will firstly present (section 2) the integration problem. Secondly, we will briefly review (section 3) the current state-of-the-art in EAI through presenting mainly two major kinds of solutions: traditional EAI systems and Web-Services-based EAI systems. Finally, we will describe (section 4), before concluding, some aspects of our work which attempts to provide a solution for the integration problem.

2. Integration Problem

In this paper we will mainly focus on Enterprise Application Systems (EAS). Typically an EAS can take many different types including batch applications, traditional applications, client/server applications, web applications, application packages [14]. These systems are often materialized in enterprise reality in form of ERP, CRM, SCM, and legacy systems.

An appropriate characterization of EASs in the context of EAI is that EASs are HADI (Heterogeneous, Autonomous, Distributed and Immutable) systems [4]:

- Heterogeneous systems mean that each EAS implements its own data and process model.
- Autonomous systems refer to the fact that each EAS runs independently of any other EAS.
- Distributed systems mean that each EAS locally implements its data model, which it generally do not share with other EASs.
- Immutable systems mean that each EAS is generally treated as black (unavailability of code and interfaces) and in best cases as gray (unavailability of code but availability of interfaces) boxes in order to access it.

The consequence of the characteristics above is that EASs are generally standalone software entities, which form what we often call islands of information or islands of automation. In this case, any form of integration of the EASs must happen outside of the involved EASs, by using integration systems such as EAI systems. This integration consists then in interconnecting the interfaces of each EAS using technologies supported by the integration systems such as queuing systems, file systems, databases or remote invocations.

The characteristics of EASs form the main reasons of the existence of the integration problem, and the more these characteristics are extremes, the more the integration become hard and complex. Despite the importance of the problems described above, we focus only, in this paper, on the heterogeneity problem, precisely the semantic heterogeneity problem, which is the hard problem of enterprise integration in general, and EAI in particular [4].

Since EASs are HADI, a semantic mediation is needed in order to achieve their integration. Its aim is to resolve all the semantic conflicts that can arise between the exchanged data, and also between invoked behavior interfaces. Indeed, data semantic mediation provides mechanisms to preserve the meaning of the data during the flow exchanges between EASs, whereas behavior semantic integration provide mechanisms to resolve semantic behavior interface heterogeneity when EASs invoke each other.

Furthermore, the integration problem is more complicated by our industrial context concerned by a large enterprise in the multidisciplinary and complex microelectronics area. This particular context is mainly characterized by several and heterogeneous business domains that needs sophisticated semantic integration in order to achieve the semantic integration.

3. Integration solutions

In this section, we will describe the major existing EAI solutions, which will be followed by some pertinent related works about EAI.

3.1. Today's EAI Solutions

Before describing today's main EAI solutions, let us remind that the fundamental EAI drivers is the existence of EASs which are HADI (heterogeneous, autonomous, distributed and immutable) systems. Generally, this is the consequence of the emergence of e-business, enterprise mergers and consolidations, and the rise of application packages and COTS (Commercial On The Shelf) [10].

In this paper, we will consider only two main existing solutions, which are the most important ones in the context of EAI: traditional EAI systems and WSs. These solutions can fulfill major integration requirements such as data synchronization, business process execution, reconciliation of technical and syntactic differences, fast deployment of new applications and so on.

3.1.1. Traditional EAI Systems

Currently, EAI systems are based on a lot of technologies such as: message brokers, process brokers, message-oriented middleware, etc. Even if EAI systems may differ from a technological point of view, the main functionalities remain the same and we can mainly distinguish five components, which provide respectively transport services, connectivity services, transformation services, distribution services and process management services [10].

The principle of EAI systems is based on using interfaces (connectors) to integrate EASs. The interfaces convert all traffic to canonical formats and protocols. These interfaces constitute the only mean to access EASs, and they can occur in different levels: user-interface level, business logic level and data level [18].

Although EAI systems address technical and syntactical integration, nevertheless they must address the semantic level which is more difficult and which can provide more added value. Today, no traditional EAI system can provide mechanism that correctly supports semantics. In best cases, data is passed between EASs by-value, and in general no shared semantic concepts are explicitly used to define semantics through different messages or to semantically describe the behavior that is provided.

3.1.2. Web Services

WSs are considered as a result of convergence of Web with distributed object technologies. They are defined as an application providing data and services to other applications through the Internet [15]. WSs promote an SOA (Service-Oriented Architecture) that is based fundamentally on three roles: *service provider*, *service requestor* and *service broker*; and three basic operations: *publish*, *find* and *bind*, and any particular EAS can play any or all these roles [17].

WSs constitute the most important concretization of the SOA model. They can be deployed inside (EAI) or outside (B2B) the enterprise. In all cases, WSs are published with appropriate URLs by WS providers over the Internet or intranet. Once published, these WSs are accessible by WS consumers via standards Web such as HTTP, SOAP, WSDL and UDDI. In addition to this, WSs can be used for integrating EASs via standards such as BPEL, or WSFL.

WSs are very promising in solving the integration problem. Today, some new integration products based on WSs standards exist and will certainly replace in the near future the proprietary solutions that are the traditional EAI systems [3].

Even if WSs are promising, they do not correctly address the semantic aspect that is currently somewhat supported by UDDI registries with the help of some standard taxonomies such as NAICS, UN/SPSC and ISO 3166 [8]. In addition, WSs do not provide neither data nor behavior mediation [5][7][11]. These drawbacks are mainly due to the lack of service ontology and mediation support in current WSs. This lack penalizes the efficiency of current WS integration in the context of EAI.

3.2. Related works

Recently, the importance of WSs has been recognized and widely accepted both by industry and academic research. This section will review some important related works about enterprise integration, mainly those that concerns WS-based integration and ontology-based integration.

In the context of data integration, there are many general works which use ontology-based approaches such as COIN project [12], OBSERVER project [21], INFOSLEUTH project [34], BUSTER project [27] and so on. All these work are not concerned about the mediation in the context of SOA.

In addition to the listed related works above, there are some other works that are addressing the WS viewpoint such as Active XML from GEMO project [1] and SODIA from IBHIS project [29]. Active XML extends XML language by allowing embedding of calls to WSs. SODIA is an implementation of Federated Database System in the context of WSs. These works do not support any mediation services.

In the context of application and process integration, some important initiatives and works exist around the semantic web service concept [8] [20] that aim to bridge the current WS gap such as OWL-S [25] [31], BPEL [9], WSMF [11], SWSI [28], METEOR-S [22]. OWL-S provides an ontology markup language in order to semantically describe capabilities and proprieties of WSs. BPEL is a standard providing a language to define business processes that can be used in application integration. WSMF and SWSI are initiatives that provide frameworks in order to support the concept of semantic web service. METEOR-S is an effort, which provide semantic web services through the extension of WS standards (WSDL, UDDI). But, most of these efforts do not provide mature concepts for mediation in the context of EAI.

4. Our Approach for Application Integration

This section will succinctly describe some important characteristics of our approach called ODSOI that mainly rely on the use of ontologies and that aims to extend the state-of-the-art in EAI in order to address the semantic problem.

4.1. Global Architecture

First of all, ODSOI approach is a solution to the information system integration problem. This means that our approach addresses the heterogeneity problem by providing a mediation-based solution using ontology concept. Indeed, our approach is based on service-oriented since it uses WSs for integrating EASs. The architecture integration that we suggest is called ODSOA (ODSO Architecture). This latter extends SOA with a semantic layer that aims to enhance service mediation in the context of EAI.

The ODSOA concept provides a unified framework in order to integrate EASs. In this framework, three main types of services (*Fundamental-Services*) are defined: *Data-Services*, *Functional-Services* and *Business-Services*. This different types can respectively address data, application and process integration.

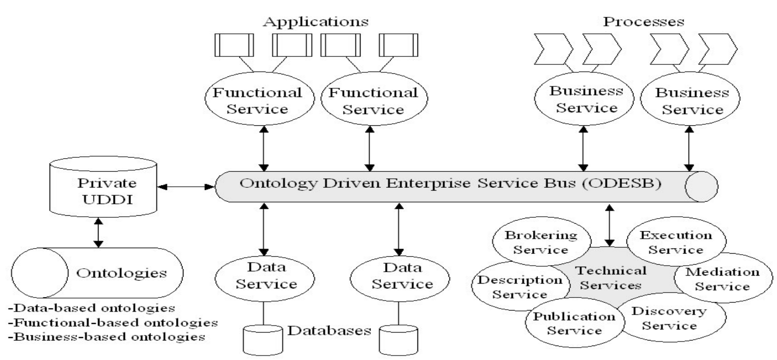


Figure 1. Global View of ODSOA Architecture

Data-Services (DS) are services that expose data sources as a service. *Functional-Services* (FS) are services that expose application systems, fundamentally functional systems (software that can perform enterprise functions such as administrative and technical ones). *Business-Services* (BS) are defined as the combination of the above services in order to expose business processes. Our service typology can be seen as an extension of the one proposed by [29] which distinguishes two concepts: SaaS (Software-as-a-Service) and DaaS (Data-as-a-Service).

Figure 1, which is a particular SOA, recapitulates these important types of services. Indeed, there are of course some other important technical services that are mainly *Brokering-Services*, *Description-Services*, *Mediation-Services*, *Publication-Services*, *Discovery-Services* and *Execution-Services*. Some of them will be described below.

A cross section of the integration bus (also called ODESB – Ontology-Driven Enterprise Service Bus) (figure 2) shows many concentric existing standard layers such as *Transport layer, Exchange layer, Registry layer* and *Transversal layer*.

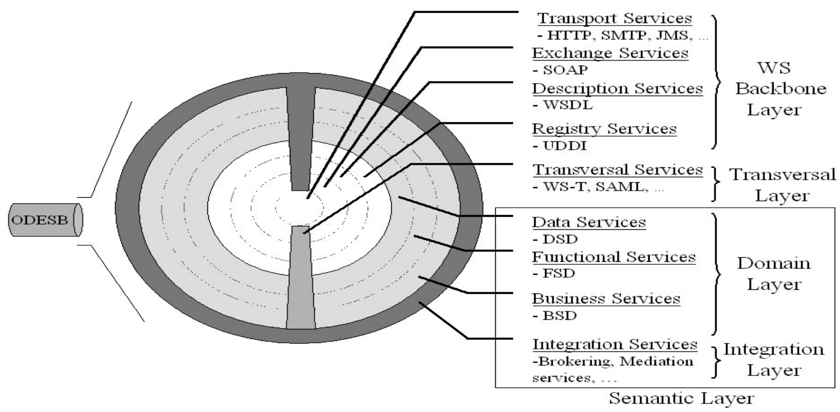


Figure 2. Cross Section of the ODESB Bus

In addition to these standard and existing layers, we suggest to adopt in a similar way as semantic web services, another layer, called *Semantic-Layer*, which includes two sub-layers that are *Domain-Layer* and *Integration-Layer*. The *Domain-Layer* aims to describe and publish the three fundamental services described above using specific descriptions such as DSD (Data Service Description) for DSs, FSD (Functional Service Description) for FSs, and BSD (Business Service Description) for BSs. All these descriptions exploit some specific ontologies and are the specialization of OWL-S (Web Ontology Language-Services). Concerning the *Integration-Layer,* it provides some *Technical-Services* in order to semantically discovery, mediate and execute fundamental services that are described and published by the layer above. In the next section, some important *Technical-Services* of the *Semantic-Layer* will be developed.

4.2. Semantic Layer Services

Semantic-Layer services are the main services that address the semantic problem.

They are divided into *Domain-Layer* services and *Integration-Layer* services. The most important technical service of each layer (which are *Description-Services* and *Mediation-Services)* will be described below.

4.2.1. Description Services

The principle of ODSOA is based on the use of some knowledge registries that store some formal ontologies, which are exploited by *Description-Services* in order to define the semantic description of services. According to Gruber, an ontology is defined as an explicit and formal specification of a conceptualization [13], and for our purpose, we have defined three major types of ontologies: information or data-based ontologies, behavior or functional–based ontologies and process or business-based ontologies.

Data-based ontologies are the most basic ones. They provide semantic description of the data. Theses ontologies are required in all cases, no matter if we leverage functional-based or business-based ontologies.

Functional-based ontologies define semantic description around functions that are provided by the multiple EASs (and then services) and that can be remotely invoked. These ontologies are generally required in order to provide a better reuse of functionalities.

Business-based ontologies define semantic description around coordinating business processes. These ontologies are generally required in order to integrate business processes.

Furthermore, *Description-Services* are based on the context of a service (*Service-Context*), which is defined by a set of ontologies, related to the concerned service. This service-context is also called local ontology, which means that there are several ontology levels. Within the microelectronics area, and precisely in the case of microelectronics society, three ontology levels have been identified: local level, domain level and global level (figure 3).

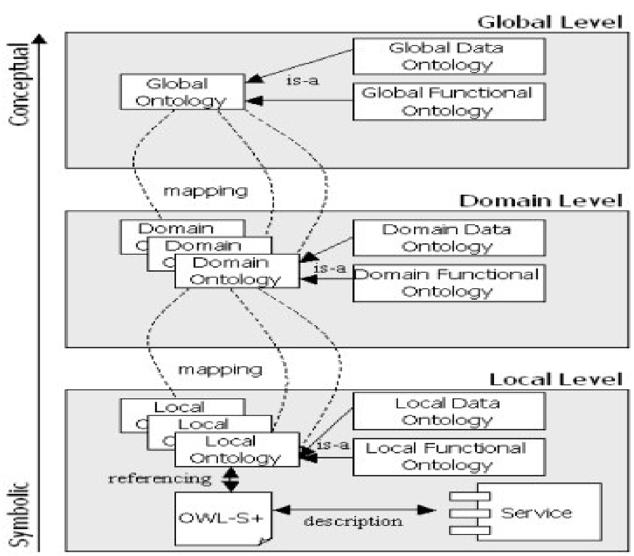


Figure 3. Excerpt of our three-Level Architecture Ontology

In essence, local ontologies concerns services, whereas domain ontologies concern the generalization of local ones that belong to the same domain (Production, Metrology, Packaging, etc.) and they can serve in aligning the involved local ontologies. At last, global ontology is considered as generalization of domain ontologies, it is the root of the ontology hierarchy, and they can serve both in aligning domain ontologies and in B2B integration that constitutes a natural prospect of our work.

Our ontology architecture is somewhat an extension of the hybrid ontology approach mentioned in the case of information integration in [32]. This extension is motivated by the fact that none of the approaches proposed by [32] (single ontology, multi-independent-ontologies and hybrid-ontology approach) are appropriate to fully capture and correctly structure semantics in our case.

This ontology clustering, which is firstly used in a general fashion in [30], is a very important concept in order to master the ontology evolution. We call this structuring *Ontology Urbanization.* It takes an important role in our integration approach and it will be more developed in future work.

4.2.2. Mediation Services

Mediation-Services are generally invoked by *Brokering-Services* (technical services that aim to provide global mechanism to integration process) in order to perform matching or resolution of semantic heterogeneity between. They exploit the description provided by the *Description-Services* described above.

Since we use an hybrid ontology approach, this requires the integration (mediation) of ontologies which are performed by *Ontology-Mediation-Services* (OMS) and that are based on ontology mapping [16]. This latter is the process whereby two or more ontologies are semantically related at conceptual level. According to the semantic relations defined in the mappings, source ontology instances can then be transformed (or matched with) into target ones [23].

In addition to OMS and according to the above different fundamental types of services, we can mainly distinguish three other types of mediation services: Data Mediation Service (DMS), Functional Mediation Service (FMS), Business Mediation Service (BMS). These mediation services aim to mediate respectively between DSs, FSs, BSs and they are based on OMS that matches and mediates between different ontologies. To be performed, mediation-services can exploit two particular utility services that are inference service and matching service. These particular services can be respectively supported by academic or commercial inference engine and matching tool. For the initial prototype that is ongoing, we decide to use Racer engine [26] and OLA (OWL Lite Alignment) matcher [24] that seems be appropriate to our approach.

4.3. Generic Integration Process

While the main components of our approach are enumerated, let's give now the generic scenario of the integration process. As shown on figure 4, it starts once the WSs have been described (semantic description) by using description services. After that, they are published (semantic publication) both in a specific service ontology registries and in a private UDDI registry (by publishing services) and then they can be discovered (semantic discovery) by discovery services in order to carry out the realization of a given task modeled as a service query (that corresponds to a user or brokering-service query) by the integration service. The discovery service can use mediation service in order to perform the semantic matching. The invoked mediation services exploit a similarity function that calculate rapprochement between ontology concepts and then between the involved characteristics of services. Once the desired WSs have been discovered, they are mediated in order to resolve the semantic heterogeneity differences by other types of mediation services (DMS, FMS, BMS). Finally, the mediated services are executed by the execution service and can invoke the integration service which can then perform another similar loop of the integration process.

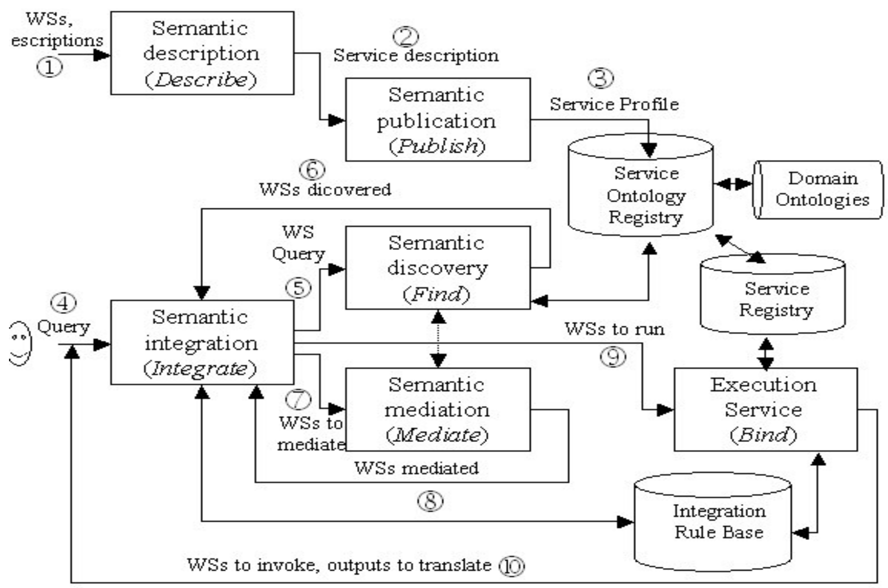


Figure 4. Generic integration process synopsis

4.4. The initial Prototype

The initial prototype (also called ODSODI - Ontology-Driven Service-Oriented Data Integration) [14], which is ongoing, aims to provide a first implementation of some functionalities of our architecture. We have restricted this first prototype to data integration. Further versions of the prototype will address application and process integration.

The underlying architecture of this first prototype is based around a fusion of WS concepts with the concepts of data mediation, especially the mediators concepts like those defined by [33].

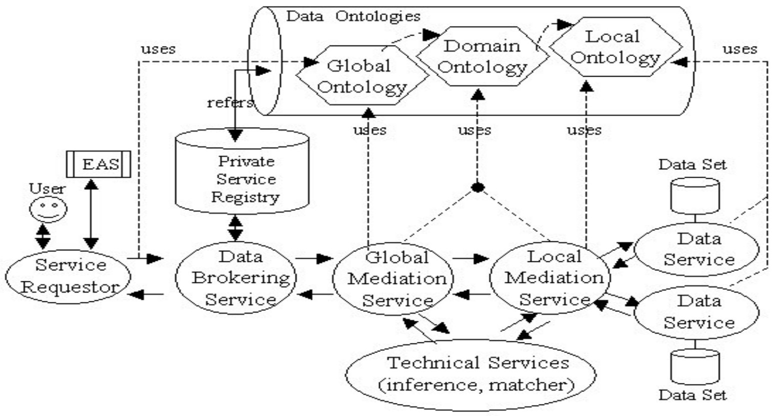


Figure 5. Principle of the initial prototype (ODSODI)

Figure 5 illustrates the principle of the ODSODI prototype. As shown, this prototype implements a local-centric approach (aka local-as-view approach) [6]. In this approach, the query is done over the global ontology and the mediation service access the data sets by a series of mappings: from global to domain (which are done by global mediation service), and then from domain to local (which are done by local mediation services). This choice is appropriate in the context of EAI in general and in the context of our microelectronics society in particular. It is motivated by the fact that users and EASs are autonomous and have a limited knowledge about the data services.

5. CONCLUSION

The semantic integration of Enterprise Application Systems is a hard problem that can concern data, applications and processes. This problem needs ontology-based semantic mediation and is best resolved in the context of service-oriented architectures.

This paper has focused on proposing a unified approach for Enterprise Application Integration that exploits both ontology mediation and Web services. This approach called ODSOI (Ontology-Driven Service-Oriented Integration) aims to extend the current web services stack technology by a semantic layer offering semantic services that can mainly define the service semantics and also perform semantic mediation in the context of EAI. Typologies of services and also of ontologies have been suggested, and the initial prototype is described. This latter is of course limited, and its extensions, which may increase the field of use and the usefulness of our approach, will no doubt constitute important prospects in the future. The further coming works will detail the implementation aspect of the proposed prototype and then extend the latter in order to address both application and process integration in the context of EAI.

References

1. Abiteboul S., Benjelloum O., Milo T., 2002. Web Services and Data Integration. *INRIA report*.
2. Amjad U., 2004. The Emerging Role of the Web for Enterprise, *Proceedings of the IEEE, 92(9), pp. 1420-1438*.
3. BIJOnline, 2004. www.bijonline.com.
4. Bussler C., 2003. The Role of Semantic Web Technology in EAI. *In the Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*.
5. Bussler C., 2004. Semantic Web Services. *ICWE'04*. Munich.
6. Calvanese D., De Giacomo G., Lenzerini M., 2001. Ontology of integration and integration of ontologies. *In Proceedings of the 2001 International Description Logics Workshop (DL2001), pp. 10-19*.
7. Cordoso J., Bussler C., Sheth A., Fensel D., 2002. Semantic Web Services and Processes. *Tutorial at Federated Conferences*.
8. Dogac A., 2004. Semantic Web Services, *ICWE'04*, Munich.
9. EBPML, 2004. www.ebpml.Org.
10. Erasala N., Yen D. C., Rajkumar T. M., 2003. Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces 25, pp. 69-82*. Elsevier.
11. Fensel D., Bussler C., 2002. The Web Service Modeling Framework WSMF. *In Electronic Commerce Research and Applications, 1(2)*. Elsevier.
12. Goh C. H., 1997. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. *Phd Thesis, MIT*. USA.
13. Gruber R. T., 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. *KSL, Standford University*.
14. Izza S., Vincent L., Burlat P., 2005. A Unified Framework for Application Integration. *ICEIS' 05 - USA*.
15. Kadima H., Monfort V., 2003. Les Web Services. *Dunod*.
16. Kalfoglou Y., Schorlemmer M., 2003. Ontology Mapping. *The Knowledge Engineering Review, Vol. 18(1), pp. 1-31*. Cambridge University Press.
17. Kontogiannis K., Smith D., O'Brien L., 2002. On the Role of Services in EAI. *In the Proceedings of the 10th International Workshop on Software Technology and Engineering*.
18. Lithicum D. S., 1999. Enterprise Application Integration, *Addison-Wesley*.
19. Linthicum D. S., 2004. Next Generation Application Integration. *Addison-Wesley*.
20. McIlraith S., Son T., Zeng H., 2001. Semantic Web Services, *IEEE Intelligent Systems, pp. 46-53*.
21. Mena E., Kashyap V., Sheth A., Illarramendi A., 1996. OBSERVER. *In Proceedings 1st IFCIS International Conference on Cooperative Information Systems, Brussels*.
22. METEOR-S, 2004. http://lsdis.cs.uga.edu/Projects/ METEOR-S.
23. Noy N. F., 2004. Semantic Integration: A Survey Of Ontology- Based Approaches. *Standford Medical Informatics. Standford*.
24. OLA, 2004. www.iro.umontreal.ca/~owlola/alignment. html.
25. OWL-S, 2004. www.daml.org/services/owl-s/1.0/.

26. Racer, 2004. http://www.racer-systems.info.
27. Stuckenschmidt H., Wache H., Vogele T., Visser U, 2000. Enabling technologies for interoperability. *Workshop on the 14th ISCSEP, pp. 35-46. Germany*.
28. SWSI. 2004. Semantic Web Services Initiative, www.swsi.org.
29. Turner M., Zhu F., Kotsiopoulos I., Russel M., Bennett K., Brereton P., Keane J., Rigby M., 2004. Using Web Service Technologies to Create an Information Broker. *26th International Conference on Software Engineering (ICSE'04). IEEE Computer Society. pp 552-561*.
30. Visser P., Tamma V., 1999. An experience with ontology clustering for information integration. *In Proceedings of the IJCAI-99 Workshop on Intelligent Information Integration, Sweden*.
31. W3C, 2004. www.w3.org.
32. Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., 2001. Ontology-Based Integration of Information. *In Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing (2001) pp.108-117.*
33. Wiederhold G., 1992. Mediators in the Architecture of Future Information Systems, *IEEE Computer Society, 25(3), pp. 38-49*.
34. Woelk D., Tomlinson C., 1994. The InfoSleuth project. *In Second World Wide Web Conference '94: Mosaic and the Web*.

Institutional Information Systems engineering: the laws based approach

Abdelaziz Khadraoui

Matis Geneva Team – University of Geneva
Uni-Dufour, rue du Général Dufour 24, CH-1211 Genève 4, Suisse
Phone +41 22 379 76 70 / Fax: +41 22 379 77 80
abdelaziz.khadraoui@cui.unige.ch

Abstract. Our research proposes a framework for Institutional Information Systems (IIS) engineering based on the laws, therefore, called the laws based approach. In this framework, the legal texts are exploited as main source of knowledge for IIS engineering. Legal texts describe in a precise way concepts, rules and constraints governing the given institutional domain. The exploitation of these sources of knowledge permits to enhance IS adequacy and compatibility with institution activities and to find stable common information for IS engineering in perspective of sustainable development.

1 Context and Motivation

Nowadays, more and more activities of the real world are supported by the Information Systems (IS). Therefore, IS constitutes one of the most critical components of an institution. In order to be conformed to the changes in their environment, these institutions are constrained to rebuild the pre-existing IS. Several research and industrial projects have addressed these specific problems of concerning IS reengineering and IS evolution. In this context, several research and industrial efforts concentrated on creating new and more efficient development methodologies and processes in order to support intensively the institution activities of the persons who are in charge of tasks and decisions.

The institution activities are governed by legal texts that regulate their execution. Thus, the contents of legal texts for the institutional domain are indispensable. Therefore, these legal texts are considered as a common universe of discourse for all the persons concerned by the IS. An IS is continuously modified due to the evolution of informatics environment as well as the modifications of its organisational environment. In this movement, it's important to have stable components. The legal texts are stable in this movement. In other words, the exploitation of these legal texts permits to find stable common information, which is necessary for institutional activities, and which should be used in the perspective of IS engineering. This information is used to form the kernel of the IS. This kernel can evolve to integrate the organizational aspects, which are not described in the legal texts.

2 The laws based approach

Our research domain aims at providing new approach for IS engineering. This approach allows to clarify the links between laws and IS, in particular the alignment between the amendment of laws and the evolution of IS. The evolution of laws is a remarkable phenomena outside of the IS that may provoke the evolution of IS. In this case, the evolution of the laws is considered as an event. We consider three types of event :(i) amendment of a law, (ii) abrogation of a law and (iii) introduction of a new law. After each legislative event, it is important to identify the parts of IS, which require an evolution. The impact of this change on IS is precisely identified.

In our research project, we apply the method components [6] to develop the institutional IS. These method components aim at constructing four levels of IS infrastructure: (i) the level of IS ontology, (ii) the level of IS kernel, (iii) the level of IS implementation, and (iv) the level of IS activity modelling. These different levels are not isolated, they can be coordinate to support the activities at the global level.

The overall view of proposed approach for IIS engineering is synthesised by the figure 1:

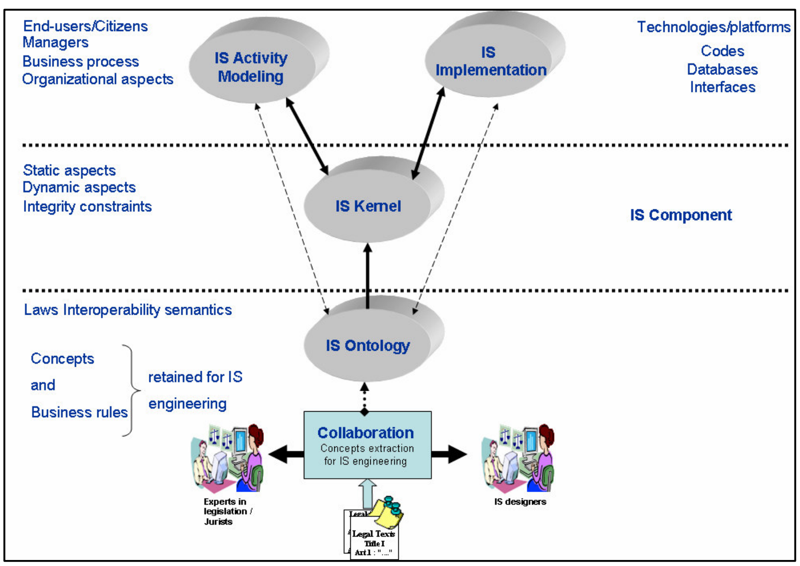


Fig. 1. IIS engineering.

In our standpoint, the level of IS ontology play the most important level. This level will conduct the construction of other levels. For this reason, our research concerns with the construction of IS ontology.

Providing indisputable “universe of discourse” based on legal texts is the first purpose of these method components. For a given domain, we propose firstly, to identify the set of legal texts such as laws, and application regulation, etc. which formalize the institutional domain. A domain can concern several legal texts. The aim of this step is to extract the IS ontology model. It defines the concepts that describe the knowledge contained in the legal texts, which can be exploited at the activity modelling level and at the level of implementation through the level of IS kernel. The study of each of these texts should be made only in the perspective of IS engineering. The question to answer is: ‘How to determine in the legal texts the IS parts, which have to be modified and how to determine the business rules contained in the texts of law’. The legal texts contain information and knowledge of purely legal nature, which cannot be considered in the IS. Only the characteristic concepts of the domain are identified and retained.

IS ontology model provides a reference for IS designers to discuss and understand the ways in which they view and interpret the institutional domain. It allows to conduct the evolution of the IS, which have been adapted to the evolution of the legal texts. IS ontology level remains independent from technologies and business practices.

The analysis and interpretation of certain laws is really a complex process. In fact, an important effort is required to carry out this process. Therefore, the collaboration between experts in legislation and IS designers is necessary for this task. We developed a first prototype of a collaborative platform to support their tasks of extracting relevant concepts from laws. This prototype of a collaborative platform has put into practice. At present, we are continuing our work on improving the protocol of collaboration between IS designers and experts in legislation.

Besides, another interesting question is how to structure this “universe of discourse” in order to establish indisputable IS kernel. IS engineering requires the use of models to support the rigour and the communicability between the various actors and levels.

In our work, we use a conceptual graph namely *conceptual map* to represent the IS ontology model [3], [7]. A conceptual map is an oriented graph in which the nodes are the concepts and the edges between concepts are either (i) instantiations or (ii) existential dependencies or (iii) generalization-specialization links.

The main characteristics of our conceptual map are:

- It does not distinguish the static aspect from dynamic aspect (static-dynamic integration). These choices concern the implementation level and not the ontological level.
- It uses only existential links, generalization-specialization links and instantiation links. These links are easy to understand; they do not introduce ambiguity and can be directly implemented. These links permits to describe the remarkable situations contained in the legal texts.

As mentioned above, the IS kernel is derived from the IS ontology model. For this reason, the mapping guidelines must be defined. The derivation process asks the IS designers to consider the IS with an operational optic. They have to make choices and to decide for each concept and each link between concepts if it is mapped into a class, or a transaction or an attribute, and so on. An IS kernel can be a formal or semi-formal representation. The kernel can be described through several diagrams which represent different aspects of IS such as the static aspects, the dynamic aspects and the integrity rules aspects. It represents the information, which is handled, used and exchanged by the activities of the institutional domain.

References

1. Boer, A., Van Engers, T.: A Knowledge Engineering Approach to Comparing Legislation. Knowledge Management in e-Government KMGov-2003, Greece, May 2003.

2. Chutimaskul, W.: e-Government Analysis and Modeling. Knowledge Management in e-Government KMGov-2002, 3rd International Workshop on "Knowledge Management in e-Government" jointly organised by IFIP WG 8.3 & WG 8.5, GI FA 6.2, University of Linz and University of Roskilde, 23 – 24 May 2002, Copenhagen, Denmark.

3. Khadraoui, A., Turki, S., Aïdonidis, C., Léonard, M.: Institutional Information Systems Alignment on Laws. AISTA 2004, International Conference on Advances in Intelligent Systems - Theory and Applications, In cooperation with the IEEE Computer Society, Luxembourg, November 15-18, 2004.

4. Léonard, M.: IS Engineering Getting out of Classical System Engineering. 5th International Conference on Enterprise Information Systems, École Supérieure d'Électronique de l'Ouest, Angers, France, pp. 35-45, 2003.

5. Le Dinh, T.: Information System Upon Information Systems: A Conceptual Framework. Phd Thesis, University of Geneva, November, 2004

6. Ralyté, J., Rolland, C.: An approach for method reengineering. Proceedings of the 20th International Conference on Conceptual Modeling, ER2001, Yokohama, Japan, November 27-30 2001

7. Turki, S., Aïdonidis, C., Khadraoui, A., Léonard, M.: Ontologies for Institutional IS Engineering. Open INTEROP Workshop On "Enterprise Modelling and Ontologies for Interoperability"; EMOI - INTEROP 2004; Co-located with CaiSE'04, Conference Riga (Latvia), 7-8 juin 2004.

Measuring the quality of ontology mappings: A multifaceted approach.

Jennifer Sampson

Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
Jennifer.Sampson@idi.ntnu.no

Abstract. While there is considerable research towards developing solutions and methodologies for ontology mapping, there is very little research towards evaluating the quality of both the tools and the outputs of the mapping process. Therefore the goal for this research is to develop a methodology and formal quality measures for evaluating ontology mapping results. This research will also require the definition of experimental protocols with fixed controls that will make the results of different evaluations comparable. Initial quality goals to be formally defined are syntactic, semantic, taxonomic and procedural quality.

1 Introduction

Ontologies are to provide a shared understanding of common domains, and provide a means for data exchange at a syntactic and a semantic level. Recently several ontology mapping techniques and approaches have been proposed as one step towards achieving interoperable systems. A mapping between ontologies means that for each concept (relation, property) in one ontology we try to locate a corresponding concept in the second ontology, with the same or closest meaning.

Problems with interoperating systems stem from the fact that the developers of information systems use different vocabularies to express the information contained in the systems even when describing the same domain. In many cases applications require information from multiple information sources. However, an application can only use information from two systems completely and accurately if it is able to establish the precise semantic correspondences among the concepts in the two systems. This is a very difficult problem to solve and current approaches often result in missing correspondences – whereby both matches and mismatches can produce errors of omission [10]. The implications of using incorrectly matched information in decision making are vast. [10] comments that, “for information sources using diverse vocabularies, an application that interoperates between the sources needs precise and complete articulations of the ontologies that contain the specification of those vocabularies” (p.53).

Ontology mapping is where we relate similar concepts or relations from different ontologies to each other by an equivalence relation. Ontology alignment is closely related but is where we bring together two or more ontologies into mutual agreement, with the aim of making them consistent and coherent. There are many tools which can, to differing degrees, achieve a matching via either term-based matching, structure-based matching or instance-based matching.

2 Research Purpose

The research goal is to develop a methodology and formal quality measures for evaluating ontology mapping results. One sub goal is to evaluate ontology mapping tools by determining the quality of the mapping outputs. Another research sub-goal is to define formal metrics for comparing the distance between ontologies. This work will also require the definition of experimental protocols with fixed controls that will make the results of different evaluations comparable.

To determine the quality of mapping results we may also need to determine the quality of individual ontologies before the mapping/merging process occurs. Initial quality goals to be defined are syntactic, semantic, taxonomic and procedural. In order to evaluate the quality of an ontology within these quality goals we need to know what kinds of things are there and can there be, in an ontology. For determining semantic and taxonomic quality we need to be able to formally define the correctness or otherwise of the taxonomic structure. [4] have defined Ontoclean as a methodology for ontological analysis. They propose four fundamental notions of formal ontology: identity, unity, essence and dependence, and attempt to show how they can be used as foundation of a methodology for conceptual modeling and ontology modeling. However [5] claims the formal statements of their proposals as logical schemata are flawed in many ways. Moreover [5] argues that [4] use an inconsistent notation making the intended semantics of the logic unclear, and [4] make false claims of logical consequence, and definitions that provably result in the triviality of some of their property features.

2.1 Research questions

The main research questions are:
"How can we measure the quality of an ontology mapping given we have diverse tools, and diverse user requirements which impact on the degree or type of mapping?"
How do we define practical criteria (quality characteristics) for evaluating the quality of ontologies and ontology-mappings?

2.2 Sub-questions

What is ontology-mapping and how can we classify the different approaches to mapping?
What parts of an ontology can be semantically interconnected?

Why evaluate the quality of ontology-mappings? Is it possible to evaluate the quality of ontology-mappings? Do we need to evaluate the quality of individual ontologies as a prerequisite for evaluating the quality of the ontology-mapping result?
What current tools are available for ontology-mapping, and what kinds of mapping techniques are supported by the ontology-mapping tool?
How can we measure the distance between two or more ontologies?
How can we measure how similar two ontologies are? Is this a more useful indicator than one which tells us how different two ontologies are?

2.3 Contribution

The research will define quality goals and quality metrics for evaluating ontology mapping results. To assist with defining these quality goals we are currently undertaking an indepth analysis of the ontology mapping and merging tools Chimaera and Prompt, to establish what are the existing problems ontology engineers face when using such tools. Once we have established an understanding of the problem we will consider formal theoretical frameworks and approaches as candidates for defining the quality measures and goals (e.g. psychologically motivated theories versus philosophically motivated theories e.g. BWW).

Once we have established the quality metrics we will conduct an empirical evaluation of current available tools (Chimaera [8], Prompt[12], Glue [1], OntoMerge[2], ONION [10]). These experiments will be to compare ontology mapping results using the proposed quality goals and metrics. The main contributions will be the development of a comprehensive methodology for ontology-mapping which describes how the proposed quality measures can be applied. We also hope to provide recommendations for future enhancements to existing tools and/or extension of an existing ontology tool which provides an automated mechanism for evaluating ontology-mappings.

In addition we would like to further test the quality metrics using a prototype agent-based ontology-mapping system. While we recognise that fully automatic ontology mapping is still impossible, we believe that software agents can assist humans in several aspects of ontology mapping for example finding useful concepts and relations among concepts from original data sources. Parts of ontologies, such as consistent concepts in data or concurrence of two concepts, can be discovered by software agents. The results of such discoveries however should be subjected to review of domain experts, or possibly other software agents. Therefore we claim that, in principle, it is possible for software agents to participate in the ontology mapping exercise. In addition we note that the ontological approach for the semantic web forces a specific approach namely taxonomic hierarchies, where the ontology describes what things are and not necessarily what they are used for. We need to consider whether taxonomic hierarchies are appropriate for meeting specialist objectives and relevant to users goals. We may consider the use of different types of hierarchies for web services for example – hierarchies that are more specific to application or workflow.

2.4 Algebraic semiotics

Algebraic semiotics [3] studies sign systems and morphisms in the framework of category theory. [3] has developed a mathematically precise theory of semiotics, called algebraic semiotics and has applied the mathematical fomalism to user interface design. In his research, [3] describes the notion of semiotic morphisms in the context of sign systems. A semiotic morphism transforms such a sign system (the source) into another (the target) supposed more suited to a particular use. Our claim is that ontology mapping can be seen as a semiotic morphism from one system of concepts to another. Semiotic morphisms are a way to describe the mapping, translation, interpretation and representation of concepts in one ontology to concepts in another ontology.

If we consider two ontologies O_1 and O_2 which are first mapped and then merged into one ontology O_G we can check the quality of the resulting merged ontology by investigating if the structure of the individual ontologies is preserved. [3] claims that "A good semiotic morphism should preserve as much of the structure in its source sign system as possible", but recognises that in many real world examples, it is not possible to preserve everything. [3] suggests that the structure that is preserved by semiotic morphisms provides an important way to compare their quality. Obviously a semiotic morphism $M : O_1 \rightarrow O_G$ need not be total, some concepts in O_1 may have no representation in O_G, we also know that some of the axioms, properties and relations of O_1 could be lost. We propose the use of semiotic morphisms as a way to measure the quality of a ontology merging process as available in the Prompt ontology merging environment [12]. Using algebraic semiotics it is possible to define and then determine a value for expressing the quality of two semiotic morphisms $M : O_1 \rightarrow O_G$ and $M' : O_2 \rightarrow O_G$.

3. Preliminary findings

This research is partly motivated by the fact that existing measures for assessing ontology pairs, for example Precision and Recall evaluation are insufficient or non existent. [11] describe tools for mapping, aligning, and merging ontologies as "tools that help users find similarities and differences between source ontologies". They claim that mapping tools either identify potential correspondences automatically or provide the environment for the users to find and define these correspondences, or both. However [11] point out that even though theories on how to evaluate either type of tools are not well established, there are already several frameworks for evaluating ontology development tools. [11] also examined the aspects that these frameworks compare: interoperability with other tools and the ability to import and export ontologies in different representation languages; expressiveness of the knowledge model; scalability and extensibility; availability and capabilities of inference services; usability of the tools. In our research we will consider and extend these aspects to develop a more formal, comprehensive approach.

3.1 Understanding quality measures for ontology mapping

While we recognise the importance of evaluating ontology tools our research is to more formally define specific quality goals for evaluating the results of an ontology mapping. Firstly we need to establish a consensus on what the desirable characteristics are required of an ontology-mapping. In order to create a meaningful set of evaluation criteria it is essential to determine the purpose, or purposes, the ontology-mapping is intended to serve. (We refer to conceptual model quality as a reference for ontology model quality). Until the publication of Lindland et al.'s paper in 1994 [7], research into determining quality in conceptual models had focused almost exclusively on the quality of the end product and resulted in a number of lists of desirable features and properties. [6] observe that previous lists of desirable properties for conceptual models have not provided a "systematic structure for evaluating them" and consequently propose a framework that not only "identifies major quality goals but gives the means for achieving them" (p.43).

The framework of [7] is primarily based on the linguistic concepts of syntax, semantics and pragmatics, in recognition that "modelling is essentially making statements in some language" and it is thus able to subsume all the characteristics they had listed from previous studies. They observed several significant trends:
· Many definitions are vague, complicated, or in some cases, even lacking...
· The list [of characteristics] is unstructured and the properties are partly overlapping...
· Specification properties are mixed with language and method properties...
· Some properties presuppose the existence of a design and even an implementation...
· Some goals are unrealistic, even impossible to reach" (ibid. p.43).
Their framework is intended to deal with those issues as well as making a clear distinction between the goals and the means, i.e. "by separating what you are trying to achieve...from how to achieve it" (ibid. p.42). The framework of [6] extends that of [7] by incorporating one further concept, participant knowledge of the domain, and two further dimensions, perceived semantics and social agreement.

In the context of ontology-mapping we have preliminary descriptions for quality assessment:
Syntactic quality: The relationship between the ontology mapping and the language used for the ontology modeling (syntactic correctness).
Semantic quality: The relationship between the ontology mapping and the ontology domain(s) (semantic correctness and semantic validity).
Taxonomic quality: The correctness of the semantic relations in the ontology. Due to the way in which ontologies are constructed we must consider that all ontologies represent valid viewpoints. Nonetheless, while there is no, one notion of a "correct" classification or taxonomy, we recognise that some taxonomic structures are more correct than others. Taxonomic quality may include the evaluation of the subtyping behaviour for each type rank (taxonomic correctness and taxonomic richness) .

[4] describe the ontological well-foundedness of the is-a partial-ordering relation. They discuss techniques based on the philosophical notions of identity, unity, essence, and dependence, which have been adapted to the needs of information systems design. They claim that they have demonstrated the "effectiveness of these techniques by taking real examples of poorly structured taxonomies, and revealing cases

of invalid generalization. The result of the analysis is a cleaner taxonomy that clarifies the modeler's ontological commitments"[4]. However like [5], we argue that the mathematical execution of their formalisation is problematic. In our research taxonomic quality will include determining the correctness or otherwise of 'is-a' and 'part-of' relations. We recognise that the 'is-a' construct is misused in the design of many ontologies, which potentially has large consequences for mapping ontologies based on incorrect classification systems.
Pragmatic quality: The relationship between the ontology mapping and the ontology designer's interpretation of the ontology mapping.
Procedural quality: The appropriateness of the approach followed to achieve the mapping.

3.2 Transformation – information loss or gain?

We can also examine the quality of an ontology mapping from the perspective of the information lost in the transformation or the information gained from the transformation. Missikoff [9] has defined possible interoperability clashes caused by differences in the conceptual schemas of two applications attempting to cooperate. The possible clashes are lossless clashes and lossy clashes:
• Lossless clashes, which can be solved with no loss of information. A lossless clash is where there exists a transformation that, given an information input produces an information output with a different representation but with the same semantic content. [9] comments that these clashes are quite intuitive and essentially of a syntactic nature. Examples include naming clashes, when the same information is represented by different labels; structural clashes, when information elements are grouped in a different way; and unit clashes, when a scalar value (typically an amount of money, or a distance) is represented with different units of measure.
• Lossy clashes, which include the clashes for which any conceivable transformations (in either direction) will cause a loss of information. [9] comments that typical cases are information represented at different levels of granularity, refinement, or precision. Another case of information loss is the presence, locally, of information that is not represented at all in the ontology [9].

3.3 Difficulties with evaluating ontology-mapping tools

Noy and Musen [11] point out that comparing and evaluating ontology development tools is very different from the comparison and evaluation of ontology mapping tools. While ontology-development tools have very similar inputs and desired outputs [11] ontology-mapping tools vary with respect to the precise task that they perform. While we can broadly say that ontology mapping tools are designed to find similarities and differences between source ontologies, [11] have identified three obstacles we need to consider when evaluating ontology-mapping tools. First they claim that tasks for which the mapping tools are designed, greatly. From the user's point of view the tools greatly in what tasks this analysis of similarities and differences supports. Second the different mapping tools rely on different inputs (e.g. only class hierarchies of the sources, classes and slots and value restrictions, existence of instances in each of the source ontologies, instances and that the source ontologies share a set of instances. Third, since the tasks that the mapping tools support differ considerably, the interaction between a user and a tool is very different from one tool to another [11].

4. Experimental Research Design

We will use controlled experiments to evaluate existing ontology mapping tools. Moreover we recognise that a controlled experiment provides the most effective way to evaluate the quality of ontology-mappings because:
It allows direct comparisons to be made between different ontology mapping results under controlled conditions through manipulation of experimental treatments.
It enables the ontology-mapping to be evaluated using objective and quantitative data.
It enables the ontology-mapping to be evaluated using independent participants.

4.1 Identifying ontology-mapping results for comparison

The experiments will only include those ontology-mappings that can be obtained by using ontology mapping tools which are freely accessible to researchers and practitioners alike. In addition due to the diversity of types of ontology mapping tools we need to establish certain groups of tools which are comparable. [11] follow the OntoWeb initiative who suggest a pragmatic classification of tools based on the users task requirements. These pragmatic selection criteria are: input requirements, level of user interaction, type of output and content of output. Initial tools we are considering for empirical evaluation are: Chimaera [8], Glue [1], ONION [10], Prompt (Protégé 2000 [12] plugin) and OntoMerge [2].

Conclusion

The overall research goal presented in this paper is to develop a methodology and a formal quality measures for evaluating ontologies and ontology mapping results. In this paper we have introduced the research project and described some of our preliminary ideas towards achieving the research goal. The aim is to empirically investigate the problems associated with ontology mapping and evaluate the existing solutions according to a solid theoretical framework.

Acknowledgments

This work is partially supported by the Norwegian Research Foundation in the framework of Information and Communication Technology (IKT-2010) program – the ADIS project.

References

1. Doan, A., Madhavan, J., Domingos, P., and Halevy A.: Learning to map between ontologies on the semantic web. In The Eleventh International WWW Conference, Hawaii, (2002).
2. Dou, D, McDermott, D and Qi, P.: Ontology Translation on the Semantic Web. In Proc. Int'l Conf. on Ontologies, Databases and Applications of SEmantics (ODBASE2003). LNCS 2888, pp. 952-969, (2003).
3. Goguen, J.: An Introduction to Algebraic Semiotics, with Application to User Interface Design. In: Computation for Metaphor, Analogy and Agents, edited by Nehaniv, C. Springer Lecture Notes in Artificial Intelligence, volume 1562, pages 242-291 (1999).
4. Guarino, N. and C. Welty, C.: Ontological analysis of taxonomic relationships. In A. Laender and V. Storey, editors, ER-2000: The 19th International Conference on Conceptual Modeling. Springer-Verlag, 2000.
5. Kaplan, A.: Towards a Consistent Logical Framework for Ontological Analysis. In Proceedings of the International Conference on Formal Ontology in Information Systems, Ogunquit, Maine, October, (2001).
6. Krogstie, J., Lindland O.I and Sindre, G.: Towards a deeper understanding of Quality in Requirements Engineering, Proceedings of 7th CaiSE, Jyvaskyla, Finland. June (1995).
7. Lindland, O.I., Sindre, G and Sølvberg, A.: Understanding Quality in Conceptual Modelling, IEEE Software, March: 42-49 (1994).
8. McGuinness, D., Fikes, R., Rice, J. and Wilder, S.: An environment for merging and testing large ontologies. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference (KR2000). Morgan Kaufmann Publishers, San Francisco, CA, 2000.
9. Missikoff, M.: Harmonise: An Ontology-Based Approach for Semantic Interoperability. ERCIM News, No. 51, October (2002) (Online edition).
10. Mitra, P.: An Algebraic Framework for the Interoperation of Ontologies. PhD Dissertation, Stanford University, August (2004).
11. Noy, N. and Musen, M.: Evaluating Ontology Mapping Tools: Requirement and Experience, In the Proceedings of the Workshop on Evaluation of Ontology Tools at EKAW'02 (EON2002), Siguenza, Spain, October (2002).
12. Noy, N. and Musen, M.: PROMPT: Algorithm and tool for automated ontology merging and alignment. In Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, (2000).
13. Protege. The Prot´eg´e project, http://protege.stanford.edu, 2002.
14. Pustejovsky, J.: Type Construction and the Logic of Constructs, In: The Syntax of Word Meaning. Bouillon, P. and Busa F. (eds.) Cambridge University Press (2001).

Integration of Business Applications using Semantic Web Technologies

Nenad Anicic[1] , Zoran Marjanovic[1]

[1]Faculty of Organizational Sciences, 154 Jove Ilica Street
11000 Belgrade, Serbia and Montenegro

1 Problem Statement

Success of large-scale, industry-wide enterprise integration efforts depends on the enterprise application integration (EAI) standards. Examples of such EAI standards include Open Applications Group (OAGIS) [1], RosettaNet [2], and Universal Business Languages (UBL) [3]. Currently, these standards are based on XML specifications that are syntactic formalisms. Capabilities of these standards and testability of integration results based on these standards are significantly limited as a consequence of the limited reasoning capabilities supported by syntactic formalisms. This follows from the fact that syntax-based approaches to define structure of business documents do not impose a common interpretation of the data and there is no way to achieve a repeatable and a verifiable procedure to recognize a semantic unit from a domain of interest [4].

Take, for example, the Schematron rules that are typically used to encode constraints for the application content of the messages exchanged among applications. These rules, however, cannot be reasoned about and compared in a context of some integration problem. Consequently, two rules that are perfectly valid syntactically may be conflicting with each other within a certain integration context.

The advent of Semantic Web offers opportunities for more capable EAI standards to capture and manipulate semantic relationships. Semantic formalisms at the foundation of these technologies allow use of computational approaches to reason about formally expressed concepts and make inferences that are useful, yet beyond the capabilities of the syntax-based approaches. Consequently, testability of the application integration efforts may become equally more powerful. Essentially, the reasoning methods, such as satisfiability and consistency checking, may be readily used to perform various types of validations, such as whether two ontologies are compatible and whether a specific business document instance has sufficient and necessary data to belong to a specific class of documents.

In principle, the Semantic Web technologies today enable one to draw automated inferences about relationships between conceptual structures using a subset of the First Order Logic formalism called Description Logics. As an example, it is possible to express constraints on existence of an element in a document schema (e.g., 'The access rights element will appear only if the sensitivity type element appears') and to reason about possible conflicts of such a rule with other document rules (e.g., 'Either the access right or sensitivity type element, but not both, will appear'). These types of reasoning are not possible using purely syntactic approaches.

2 An Approach

This effort develops an approach to evaluate capabilities of the Semantic Web technologies for EAI and, particularly, how it affects integration testing capabilities. The specific objectives that drive this work are (1) to develop an experimental tool enabling assessment of Semantic Web technologies for EAI and (2) to design and execute a series of experiments to effectively perform such an assessment. To accomplish these objectives, the work posits Semantic Web-based integration architecture and an integration methodology that is enabled by such architecture. In particular, we are interested to investigate possible advances in testability of integration efforts using the new technologies. This novel integration methodology includes a collection of integration and validation steps that are performed both at design time and run time of an integration process. During design time, the methodology supports development of generalized and normalized ontologies (that describe application interface models) and allow model-based similarity analysis of these ontological models. During run time, the methodology enables semantic translation of instances of business documents (conforming to the developed ontologies) using the previously developed ontologies and automated reasoning tools.

Initial experimental results in testing the methodology show interesting capabilities such as the ability to perform individual equivalence test that is content based. Through experimental work, we have also gained a significant number of insights into the issues of necessary and sufficient conditions for achieving interoperable data exchange.

Our future work will focus on experimental assessment of the initial ideas for Semantic Web-based EAI standards. We expect to identify key technical issues for the proposed approach, and through experimental demonstration show how such issue may be addressed by extending the proposed approach. Our key contribution, we anticipate, will be to increase significantly understanding of whether and how Semantic Web technologies may be applied in a near future to realistic industrial integration efforts.

References

1. Open Applications Group (OAG) Web Site, [Online]. http://www.openapplications.org/ , (current November 2004)
2. RosettaNet Web Site, [Online]. Available: http://www.reosettanet.org , (current Nov. 2004)
3. Meadows, B., Seaburg, L. (eds.): Universal Business Language 1.0, [Online]. Available: http://docs.oasis-open.org/ubl/cd-UBL-1.0/, (current November 2004)
4. Decker, S., Harmelen, F., Broekstra, et al.: The Semantic Web - on the Roles of XML and RDF . In: IEEE Internet Computing, Vol. 4,No. 5,63-74, (2000)

Towards a CAME Tools for Situational Method Engineering

Nicolas Arni-Bloch

University of Geneva, CUI, 24 rue General Dufour,
CH-1211 Geneva 4, Switzerland
nicolas.arni-bloch@cui.unige.ch

Abstract. While the theory of Situational Method Engineering (SME) becomes increasingly solid, very few engineering tools have been developed to support the application of its research results. In this paper we analyse the requirements for such a tool and detect the capabilities that are not yet provided by existing tools. The paper focuses on the role of the method process enactment mechanism which is generally omitted in such kind of tools. It guides the way to use the method in order to accomplish the development of corresponding schemas.

1 Introduction

Information Systems (IS) are increasingly used in all areas of our society. While the use of IS in organisations grows, the complexity and the diversity of its application domains increases as well. To address this complexity methods are used to guide Information Systems Development (ISD). However, ISD methods are often too generic and cannot be followed literally, they need to be adapted to the specific situation of each ISD project. Situational Method Engineering (SME) aims to resolve this problem and to provide techniques allowing to construct project-specific methods 'on the fly' [1]. It focuses of the formalization of methods in terms of reusable method components and the definition of assembly techniques allowing to construct new methods by reusing these components.

The aim of our work is to propose a tool supporting assembly-based SME and allowing to satisfy method requirements of each specific ISD project. In this paper we present our first steps towards such a tool.

In the next section we present an overview of SME domain and define its basic terms. In section 3 the requirements for method process enactment is discussed, while different tools necessary to support SME are introduced in section 4. The interoperability perspective in SME is described in section 5 and finally section 6 closes this paper with the discussion about our future works.

2 Overview of Situational Methods Engineering Domain

In this section we present an overview of Situational Method Engineering (SME) domain. We define here the main concepts of this domain such as method, situational method, method chunk and assembly based SME. Finally, we present the different level of the method engineering domain.

2.1 Definitions

Method. Generally speaking, a method describes a regular and systematic way how to accomplishing something. In the domain of Information Systems engineering, Brinkkemper defines a method as "an approach to perform a systems development project, based on a specific way of thinking, consisting of directions and rule, structured in a systematic way in development activities with corresponding development product" [2]. In more structured way, a method is made up of a product part and a process part. The product part represents the concepts that are used in the method, relationships between these concepts as well as constraints that they have to satisfy. The process part represents the way to accomplish the development of the corresponding product.

Situational Method. We call situational method the method which is constructed for a specific project in order to perfectly satisfy its specific situation.

Method Chunk. A method chunk is a cohesive, autonomous and interoperable part of method [3]. It is a basic building block in the development of situational methods. Method chunks are stored in a repository also called method base [2]. The notion of a method chunk descriptor is proposed in order to support method chunks selection and retrieval process. The descriptor allows to specify the reuse context of the method chunks and to classify and position them in the repository.

Assembly Technique for SME. It is clear that traditional method construction techniques cannot be applied here as they would be too expensive and time consuming. We need techniques allowing to construct new methods in a fast and effortless manner. One of these techniques is called assembly technique [4]. The method engineering process following this technique consists in three steps: first of all we have to specify the situation of the project at hand and to define its methodological requirements. Second, we have to select method chunks satisfying these requirements and finally, we have to assemble the selected method chunks in order to obtain a coherent situational method**.**

2.2 The Three Levels of Method Engineering

We distinguish three method knowledge levels in the domain method engineering. The highest level represents the *meta-knowledge* of method engineering that is the method meta-model. In this level the product and the process models of methods are specified in terms of product and process meta-models. Depending of the semantic expression capability of this level and is extensibility [5], the meta-model will be able to specify a large or small range of method processes and products. Particularly, the formality level of the process model will define the capability for a tool to realize the enactment of the method. The method enactment is discussed in the next section.

The middle level, called *method knowledge* level, specifies the method product and process perspectives in terms of corresponding models. By instantiating the method meta-model, the process model of a specific method is defined, and the product model used in this method is specified. The construction of the method hold in this level, it can be done from scratch, by assembly of method chunks or by modification of existing methods.

Once a method is specified we can use it; this occurs in the lowest level called *method enactment* level. In this level the method process model is executed and product models are instanced.

3 Enactment of the Method Process

While the product part of a method has been well defined in the literature and had a lot of tools to express it, the process part is less developed [6]. Several languages are available to define the product part of a method. Some of them are specific to a kind of product like UML for object specification, ER for database specification, other are more generic like the meta-language MetaEdit+ [7]. All these languages are supported by tools allowing to manipulate them like CASE tools (Rational rose, PowerDesigner for UML) or metaCASE tools as MetaEdit+. Unfortunately none of these tools can express the process part of a method and support the enactment of the method process model. By 'express the process part of a method' we mean the ability to specify, with a *meta-model*, the tasks and procedures and their ordering in order to achieve the method goal. By 'enactment of the method process model' we mean the ability of a tool to support the development and elaboration of a corresponding product according to the method process specification. In order to achieve the enactment of a method with a tool, the method meta-model needs to be formal and precise enough; only textual definition of a method is not sufficient to realise the enactment mechanism. The minimal requirements for a tool supporting method enactment are as follows:

Product space. The product space defines all kinds of products that will be used during the method process.

Input product parts. The input product parts represent the objects from the product space that are used by the method in order to realize the process.

Output product parts. The output product parts are the objects from the product space produced by the method by executing its process.

Body of the method. The body of the method is the description of its process; it can be a textual description, an executable action or a more complex formal or semi-formal specification of a process. A textual body describes the process part of a method in natural language. In this case, the tool supporting method enactment can just show the textual information and offer some facilities for the manipulation of the corresponding products. These facilities depend of the product types. It can be a drawing capability for UML modelling, a graphical representation of data, etc. In an executable body the process is expressed as an executable algorithm in a programming language. The method enactment tool has to be able to execute this piece of code. Finally, a complex body is a method process that is composed of other method processes. In this case the tool has to route the enactment in the diferent sub-processes.

4 Tools to Support Situational Method Engineering

Different kinds of tools are needed to support the engineering of situational methods. The first tool is a methods repository also called methods base. In this tool method chunks are stored together with their descriptors. The biggest challenge of this tool is to provide a high level method chunks classification mechanism. Chunks have to be well described in order to know what the method chunk is doing without the need to look inside its specification.

The second tool is the computer-aided methods engineering tool (CAME). This tool is based on the method meta-model and it is responsible for method chunks specification, i.e. their product and process parts definition. Method chunks specification can be done “from scratch”, by assembly or by modification. In the first case product and process models of the method chunk are defined by instantiating the method meta-model used by the tool. In the second case method chunks are assembled in order to satisfy some specific situation. In the third case method chunks are obtained by modification of other method chunks in order to better satisfy the method goal. Depending to the method meta-model, the CAME tool should offer graphical modelling facilities and special features.

The last tool is the method instantiation tool. This tool or this family of tools have to be able to perform the enactment of the constructed method. In other words, the tool has to support the development and elaboration of products according to the method process specification. It has to guide the method user in the application of a selected method chunk and to offer different features for the manipulation of the products to be constructed by using this method chunk. Therefore, the instantiation tool has to understand the process model of the method chunk and be able to execute it. In order to support product construction, this tool has to understand the product model of the corresponding method chunk and to support its instantiation.

As each ISD life-cycle uses a lot of different kind of product types, like products for the analysis phase, design, development, etc., it would be hard to have only one tool for all kinds of products. As we said before, there are a lot of tools to manipulate product parts of methods, but none of them integrates the method enactment facility. One possibility to ensure the method instantiation is to add the enactment capability to these tools. Another way is to add the enactment capability to the metaCase tools like metaEdit+. In this case we have to define the meta-capabilities for method enactment to cover a wide range of product types and activities.

5 Interoperability and Situational Methods

In this section we briefly discuss the notion of interoperability in two perspectives: (1) how situational methods and SME could help to resolve deferent IS interoperability problems and (2) what are the interoperability challenges in SME?

First of all, the repository of method chunks enables the possibility to store method chunks addressing different IS interoperability issues as for example requirements specification for interoperable systems and applications, integration of IS components, etc., These chunks can then be assembled and tuned to respond some particular interoperability situation.

Besides, SME itself has also different kinds of interoperability challenges. To enable method chunks assembly, SME approaches have to deal with interoperable method chunks and have to provide mechanisms to define differences or similarities of method chunks i.e. their product and process models. Therefore, the first challenge of SME is the method chunk definition. As we indicated before, to enable assembly based method engineering we need to have a repository containing a large number of method chunks. One of the possibilities to quickly fill the method repository of an organization is to exchange method chunks between organizations or to by them. But the question is how to ensure the compatibility between different kinds of method chunks, i.e. their process and product models? The incompatibility can be detected in different levels: classification parameters, modeling languages, process or product models specification. But this interoperability challenge is not specific to method engineering domain. It can be compared with other kind of inter-organizational exchanges such as information, data, and applications. We hope that the method chunks for IS interoperability will help to find method solutions to resolve different IS interoperability problems.

The second challenge of SME is called the tool interoperability [2]. As mentioned before in this article, there exist a lot of product types and tools supporting them that cover the ISD life-cycle. Therefore, we are confronted with the integration of these different product types and tools. This problem holds in the overall framework for SME. Different ways exist to address this challenge based on the storage structure like common repository, agent based communication or functional component in a integrate architecture.

6 Future Works

The survey of the literature shows that there are many research works made in the field of SME. Even though the theory of SME becomes more and more solid, there are very few tools available to support the specification of reusable method chunks, their storage in method repositories, their assembly and especially the enactment of methods.

In order to implement all these capabilities we have to resolve a large range of problems such as method chunks definition and classification, assembly-based method construction and method enactment. In our future perspective we consider to focus on one of these challenges, the method enactment mechanism. Therefore, we are consolidating now our method meta-model and we consider the way to implement it in a specific programming environment.

References

[1] Kumar, K. & Welke, R.J. : Method Engineering, A Proposal for Situation-specific Methodology Construction. *In Systems Analysis and Design : A Research Agenda*, Cotterman and Senn (eds), Wiley, (1992) 257-268

[2] Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. Information and Software Technology, Vol. 38. Elsevier Science B.V. (1996) 275-280

[3] Ralyté, J. & Rolland, C. (2001). An Approach for Method Reengineering. Proceedings of the 20th International Conference on Conceptual Modeling (ER2001), Yokohama, Japan, LNCS 2224, Springer (2001) 471-484

[4] Ralyté, J., Rolland, C.: An Assembly Process Model for Method Engineering. Proceedings of the 13th Conference on Advanced Information Systems Engineering (CAISE'01), Interlaken, Switzerland, LNSC 2068, Springer-Verlag (2001)

[5] Prakash, N.: Towards a Formal Definition of Methods. Requirements Eng, Vol. 2. Springer-Verlag (1997) 23-50

[6] Tolvanen, J.-P., Rossi, M., Liu, H.: Method engineering: Current research directions and implications for future research. In: IFIP TC8 Conference on Method Engineering, Chapman & Hall, 1996

[7] Kelly, S., Lyytinen, K., Rossi, M.: MetaEdit+: A Fully Configurable Multi-User and Multi-Tool CASE and CAME Environment. In: Proceedings of the 8th International Conference on Advanced Information Systems Engineering (CAiSE'96), Heraklion, Crete, Greece, Constantopoulos et al. (eds.), , LNCS 1080, Springer-Verlag (1996) 1-21

A Methodological Approach for Enterprise Modelling of Small and Medium Virtual Enterprises based on UML. Application to a Tile Virtual Enterprise

Reyes Grangel (Ph.D. Student) and Ricardo Chalmeta (Supervisor)

Dep. de Llenguatges i Sistemes Informàtics, Universitat Jaume I, 12071 Castelló, SPAIN, {grangel, rchalmet}@uji.es

Abstract. Enterprise Modelling has been used successfully for years with different purposes. Nowadays, there are a lot of languages, methodologies and tools related to Enterprise Modelling, even for modelling Virtual Enterprises. However, some of the Enterprise Modelling weaknesses have not been solved yet. One of the most important is the lack of interoperability among enterprises that use different Enterprise Modelling Languages (EML). Such EML are defined in proprietary formats, and they are only implemented by proprietary and expensive tools. So that, this problem is intensified in Small and Medium Enterprises (SMEs), because they have limited resources.

In this context, this paper shows my Ph.D. thesis proposal describing the problematic situation which is the origin of this research and the objectives suggested to solve it. The thesis goal is to investigate the possibilities of using UML 2.0 and Profiles mechanism in order to provide a methodological approach for solving interoperability problems to Small and Medium Virtual Enterprises in the context of Enterprise Modelling.

1 Introduction

The objective of this paper is to describe the proposal for my Ph.D. thesis. This document intends to give a first idea about the thesis origin and objectives. It is structured in three sections. The first one shows background and definitions related to the thesis framework. In the second one, the problematic situation that the thesis intends to solve is described. Finally, the main research objectives are presented.

2 Background and definitions

Enterprise Modelling [18] is the art of 'externalizing' enterprise knowledge, which adds value to the enterprise or can be shared, i.e., representing enterprise in terms of its organisation and operations (processes, behaviour, activities, information, objects and material flows, resources and organisation units, and system infrastructure and architectures). Therefore, this art consists of obtaining enterprise models, that are a computational representation of the structure, activities, processes, information, resources, behaviours, etc. of an enterprise, government or any another type of business. This model can be at the same time descriptive and definitional, including that what is and what should be. And its role should be to obtain a design, analysis and operation of the enterprise according to the model, i.e., driven by the model (model-driven) [11]. In conclusion, Enterprise Modelling is the set of activities or processes used to develop the different parts of an enterprise model with a definite objective.

On the other hand, Unified Modeling Language (UML) is a visual language for specifying, constructing and documenting the artifacts of systems. It is a general-purpose modelling language that can be used with all major object and component methods, and that can be applied to all application domains (e.g., health, finance, telecom, aerospace) and implementation platforms (e.g., J2EE, .NET). UML has emerged as the software industry's dominant modelling language. It has been successfully applied to a wide range of domains, ranging from health and finance to aerospace to e-commerce [16]. However, UML has been used mainly so far as a modelling language in order to produce software artifacts. Even though, some works to evaluate UML from point of view of Enterprise Modelling have been carried out by some authors [2, 9].

Moreover, the Profiles package is defined in UML 2.0 as a mechanism that allows metaclasses from existing metamodels to be extended to adapt them for different purposes. This includes the ability to tailor the UML metamodel for different platforms (such as J2EE or .NET) or domains (such as real-time or business process modeling). UML Profiles had been already defined in the previous versions of UML, but their definition has been improved in the UML 2.0, specifying better the relationships allowed among elements of the model and the use of metaclasses of a metamodel inside an UML Profile [12].

3 Problem description

Nowadays, there exist a lot of languages, methodologies and tools related to Enterprise Modelling, even for modelling Virtual or Extended Enterprises [10]. Enterprise Modelling Languages provide constructs to describe and model the people roles, operational processes and functional contents, as well as support information and production and management technologies. There exists great quantity of Enterprise Modelling Languages and they are overlapped. But the integration of the models generated with these languages is complicated, since tools do not exist to integrate models generated with different languages. In this sense, the objective is to achieve a common format, as UEML or POP*, which are valid initiatives in order to enable exchange between different models and the establishment of an environment for reusing existing models [1, 13, 14, 17].

This kind of languages are defined in proprietary formats and they are only implemented by proprietary and expensive tools. Therefore, interoperability problem is intensified in Small and Medium Enterprises (SME), who have limited resources to adapt successfully innovative technologies existing in the market.

So that, SMEs carry out few enterprise models, and moreover the exchange of them among partners is very difficult.

On the other hand, SMEs set up Virtual Enterprises in order to establish flexible collaborations with other partners and to take advantage of new market opportunities. Virtual Enterprise [3] can be define as a temporary network of independent companies, often former competitors, who come together quickly to exploit fast-changing opportunities. The business partners are integrated using information and communication technology. So, interoperability problem at different levels, including enterprise modelling level, can become decisive aspects to reach business success.

Therefore, the main problem at enterprise modelling level for Small and Medium Virtual Enterprises (SMVEs) is focused on the lack of interoperability of existing Enterprise Modelling Languages, and also on the few quantity of enterprise models generated in this kind of enterprises. However, such enterprises use UML to model and generate software artifacts. The idea of this proposal is to provide a methodological approach that can help SMVEs to use successfully UML, not only to generate software models, but also to produce enterprise models that enable them to have a holistic enterprise view and better interoperate with other partners.

4 Research objectives

The IRIS Group of Universitat Jaume I in Castelló (Spain) has been working on several projects related to Virtual Enterprise in different sectors (transport, tile industry, textile, etc.) since 1999 [4–8]. This thesis proposal is motivated inside this framework in order to improve the interoperability of this kind of enterprises at enterprise modelling level.

Therefore, the main research goal is to provide mechanisms to reduce interoperability problems at enterprise modelling level to SMVEs. In this sense, the objective is to investigate the possibilities of UML use for Enterprise Modelling in order to solve this kind of interoperability problems. Besides, the mechanism provided by UML Profiles, redefined in UML 2.0, will be analysed in order to extend and adapt UML for the specific domain of enterprise modelling in SMVEs.

The specific objectives of the research work are the following:

- To perform the state of the art in UML and UML Profiles focused on Enterprise Modelling, and in Virtual Enterprises especially in SME; taking into account the MDA [15] framework defined by OMG and European Projects related to interoperability.
- To obtain a set of requirements for modelling whole enterprise dimensions of SMVEs, in order to define a framework for describing problematic situation.
- To define a methodological approach for enterprise modelling of SMVEs based on UML, which should include the UML Profiles defined in order to extend UML for enterprise modelling, and the guidelines to use this profiles in order to generate interoperable enterprise models.

– To validate the methodological approach defined in a real case of study, applying the methodology to a Tile Virtual Enterprise.

References

1. ATHENA (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications) Project (IST-2003-2004). http://www.athena-ip.org, 2005.
2. G. Berio and M. Petit. Enterprise Modelling and the UML: (sometimes) a conflict without a case. In *Proc. of 10th ISPE International Conf. on Concurrent Engineering: Research and applications*, pages 26–30, July 2003.
3. J. Browne and J. Zhang. Extended and virtual enterprises - similarities and differences. *International Journal of Agile Management Systems*, 1/1:30–36, 1999.
4. R. Chalmeta. Virtual Transport Enterprise Integration. *Journal of Integrated Design and Process Science*, 4(4):45–55, December 2000. Publisher IOS Press publishes.
5. R. Chalmeta, C. Campos, and R. Grangel. References architectures for enterprises integration. *Journal of Systems and Software*, July 2001. Publisher Elsevier.
6. R. Chalmeta and R. Grangel. ARDIN extension for virtual enterprise integration. *Journal of Systems and Software*, Setember 2003. Publisher Elsevier.
7. R. Chalmeta and R. Grangel. Virtual Integration of Tile Industry (VITI). In *Conceptual Modeling for Novel Application Domains*, volume 2814 of the serie Lecture Notes in Computer Science. M.A. Jeusfeld and O. Pastor, October 2003.
8. R. Chalmeta and R. Grangel. Performance Measurement Systems for Virtual Enterprise Integration. *Journal Computer Integrate Manufacturing*, January 2005.
9. H.E. Eriksson and M. Penker. B*usiness* M*odeling with* UML*:* B*usiness* P*atterns at* W*ork*. J. Wiley, 2000.
10. EXTERNAL (Extended Enterprise MEthodology) Final version 1-12-d-2002-01-0 (IST-1999-10091). http://research.dnv.com/external/default.htm, 2002.
11. M. S. Fox and M. Gruninger. Enterprise Modeling. *AI Magazine*, 19(3):109–121, 1998.
12. L. Fuentes and A. Vallecillo. Una introducci´n a los perfiles UML. *Nov´tica*, marzo-abril(168):6–11, 2004.
13. IDEAS (Interoperability Development for Enterprise Application and Software) Project. http://www.ideas-roadmap.net, 2005.
14. INTEROP (Interoperability Research for Networked Enterprises Applications and Software) Project. http://interop-noe.org, 2005.
15. Object Management Group, OMG. MDA G*uide* V*ersion 1.0.1*, document number: omg/2003-06-01 edition, 2003.
16. Object Management Group, OMG. U*nified* M*odeling* L*anguage (*UML*)* S*pecification:* I*nfrastructure, version 2.0*, OMG Adopted specification ptc/03-09-15 edition, 2003.
17. UEML (Unified Enterprise Modelling Language) Project (IST-2001-34229). http://www.ueml.org, 2005.
18. F. B. Vernadat. E*nterprise* M*odeling and* I*ntegration:* P*rinciples and* A*pplications*. Chapman and Hall, 1996.

Methods Enabling Web Information Service Engineering on the Semantic Web (Extended Abstract)

Terje Wahl

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Sem Sælands vei 7-9, N-7491 Trondheim, Norway
terje.wahl@idi.ntnu.no

This paper concerns a recently started research project[1] investigating methodology issues regarding The Semantic Web (SW). The SW envisions adding structure and semantic information to the content of webpages, enabling machines to understand and reason about information on the Web in a generalized manner. So far several representation languages have been specified, including RDF for resource descriptions, OWL for specifying ontologies, OWL-S for describing Semantic Web Services and various other languages for representing business process models. Currently the research community also focuses on issues related to automated reasoners, trust and security for the SW. The areas of Semantic interoperability and usage of Software Agents for utilizing Semantic Web Services are also receiving a lot of attention.

Model-driven application development is central in Web information systems and service engineering. Many problems encountered when building applications of information systems and services involve the manipulation of models. The problem becomes more evident when the process is distributed. Moreover, during the system development process the heterogeneity of the fragments increases due to the highly interactive and iterative nature of the development process and to the different, sometimes conflicting, view of the problem and solution taken by the different stakeholders. Current SW prototypes seem to be built in an ad-hoc way. There has been a good deal of focus on representation languages for the SW, but not so much on methodology for how to use these specifications together to develop a useful solution for the users. A cohesive methodology for developing Web information systems and services is needed if the SW is to move from the research community into general use.

The overall objective of the author's thesis is to contribute to the development of methods and tools for establishing information services in the context of the SW. It is *not* a goal to develop yet another representation language, but rather to utilize existing

[1] Project 160426/431 WISEMOD (Web Information Service Modelling) is funded by the Norwegian Research Council for the period 2004-07, for more information please visit http://www.idi.ntnu.no/~guttors/wisemod/

ones to look at methods for how these can be used together to develop Web information services and systems. An integrated approach for ontology and information service engineering will be developed, including techniques for information service provision and consumption and techniques for quality management of the process and product in Web information systems development. A cohesive methodology must be developed to facilitate the coordination and management of the service development process, and a methodological approach for integration and manipulation of all information produced during the project should be developed. Web information systems and service development should also be supported by providing means for semantic interoperability and management of specification fragments independently of development phase, model perspective, view, or representation language.

Another closely related and interesting research question is adaptation and configuration of SW systems and services for non-technical users. If the SW is to attain a global reach, the author believes that it must be simple to use for non-technical business users. This means that business users must themselves be able to semantically enrich data-sources that are not automatically enriched, and perform work by composing SW Services into business processes and executing them. This will require a strong user-friendly framework as a basis for work, and a robust methodology to guide users in their work.

Existing approaches for information modeling and meta-modeling will be combined with the workflow modeling approach from the PhD-student working on Workflow Modelling (also in the WISEMOD-project) to form a comprehensive methodology for Web information service and application engineering. The result will be a new model-based methodology for the engineering of Web information services and applications, resulting in better SW applications than what can be achieved with current approaches.

So far, some research regarding state-of-the-art in SW technology has been performed. A study of issues for utilizing and enabling the global SW has been conducted. Various methodologies and their quality will be surveyed to examine what types of methods may be best suited for developing SW services and systems. This may include both methods specifically targeted to Web service development and more general software development methods. These more general methods will be surveyed by investigating to what extent they can be adapted to support Semantic Web Services development. A combination of empirical and analytical approaches will be used. A proof-of-concept prototype tool will be built to support the suggested development process and be used as a basis for practical experimentation. The practical usefulness of the suggested approaches for information service engineering will thus be evaluated. Experience reports from the various studies undertaken will form the evaluation part of the thesis.

Adapting HPM to B2M interoperability issues: Towards Interoperability between Business Management Level and Shop Floor Level

Salah Baïna [1], Hervé Panetto [1], Khalid Benali [2], Gérard Morel [1]

[1] *CRAN (UMR 7039), University Henry Poincaré Nancy I*
F 54506 Vandoeuvre les Nancy, France,
{salah.baina, herve.panetto, gerard.morel} @cran.uhp-nancy.fr

[2] *LORIA (UMR 7503), Campus scientifique,*
F 54506 Vandoeuvre les Nancy, France,
benali@loria.fr

Abstract. Enterprise-control system integration between business systems, workflow systems, manufacturing execution systems and shop-floor process-control systems remains a key issue for facilitating the deployment of plant-wide information-control systems. Since years, increasing flexibility and efficiency of their systems has been the new challenge for modern manufacturing enterprises. Business process modelling aims at specifying objects flows and processes inside enterprise levels and among networked enterprises. However, the increased complexity of these models does not help at ensuring coherent relationships between its components. Moreover, when modelling different flows could handle different views of the same objects, thus coherence have to be maintained during the whole manufacturing process. In this paper, we show how Holonic Process Models (HPM) could be adapted for Business to Manufacturing (B2M) interoperability by applying the concept of Holon. We define a Holon oriented approach and design principles for specifying and building HPMs, in the specific domain of manufacturing systems. This approach aims at increasing model abstraction in order to simplify its initial comprehensiveness and ensure coherence between different views of manipulated objects. For reusability's sake, we propose a possible interconnection of the proposed Holon model in existing enterprise models. This integration enables exchanging data between the holon based models and other existing models.

Keywords: manufacturing systems; enterprise modelling; HMS; B2M; models integration; enterprise integration; interoperability; model mapping.

1 Introduction

In the last few years, lot of work has been done in order to ensure interoperability between applications. Enterprise application integration and the opening of information systems towards integrated access have been the main motivation for the interest around systems interoperability. Enterprise Integration which consists in connecting and making interoperable all the functional areas of an organization to improve synergy within the enterprise to achieve its mission and vision in an effective and efficient manner [1]. Enterprise integration occurs when enterprise data is handled and performed in different sites, services or companies. Manipulating all this data as a whole cannot be achieved simply by connecting computers ([2], [3]).Integration aspect and information sharing in the enterprise lead to an organisation of the hierarchy of enterprises applications where interoperability is a key issue (Fig. 1). This hierarchy defines the three main levels in manufacturing enterprise:

- Process control level contains all processes that perform moves and physical transformations on the produced goods and services;

- The Execution level perform the processes that manage decision flows (Workflow) and production flows (MES);
- The management system level is responsible of the management of processes that handle all different informational aspects related to the enterprise (e.g: ERP systems).

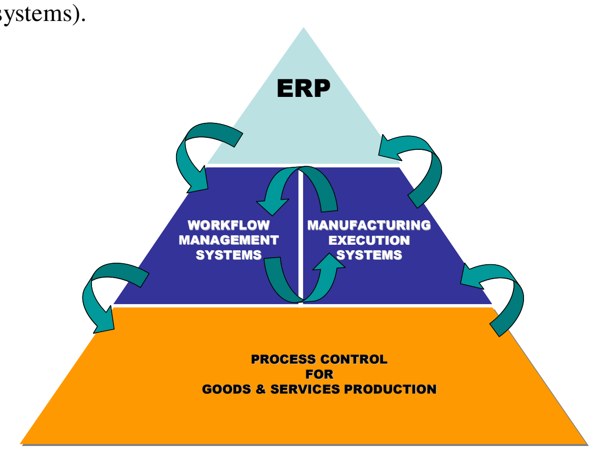


Fig. 1 Manufacturing enterprises structure

The integration of those different levels can be seen from two different points of view (Fig. 2):

- ”Horizontal Interoperability” is the interoperability between applications from the same level in the enterprise. This first category of interoperability aims to synchronise models that were created in different enterprises even those managed by different modelling systems (e.g.: enabling organisational interoperability between two systems used in two different organisations).
- ”Vertical interoperability” is the interoperability between applications from different enterprise levels. The objective of this category of interoperability is to maintain coherence between information that is handled in two different level of the enterprise (e.g.: ensuring coherence between organisational models of the enterprise and the process models used at shop floor level).

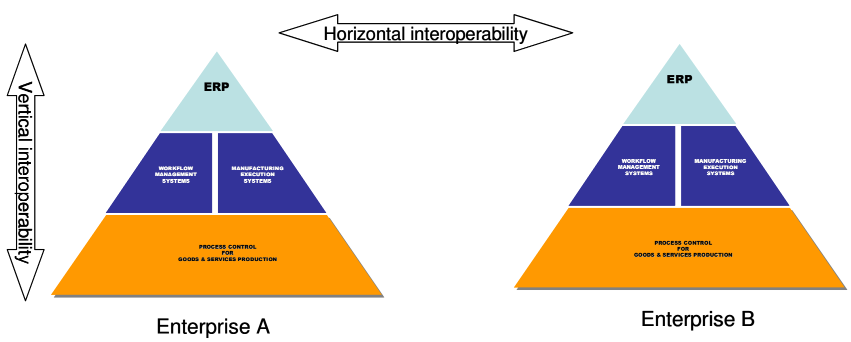


Fig. 2 Different types of enterprise interoperability

In this paper, we define a Holon oriented approach for specifying and building Holonic Process Models, in the specific domain of manufacturing systems. This approach aims at increasing manufacturing model organisation in order to simplify its comprehensiveness. We propose a modelling approach in order to handle enterprise levels interoperability concerns.

2 Applications Interoperability in Manufacturing Systems

The ISO/IEC 2382[1] Information Technology Vocabulary defines interoperability as "the capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units." The IEEE STD 610.12[2] standard defines interoperability as "the ability of two or more systems or components to exchange and use information". In this paper, interoperability definition is adapted from the two previous definitions as the ability to communicate, to cooperate and to exchange models between two or more applications despite differences in the implementation languages, the execution environments, or the models abstraction ([4], [5]). Interoperability has been studied in different domain applications. The following presents an overview of existing methods and technologies enabling interoperability.

2.1 Interoperability between ERP and MES Systems:

In this paragraph, we focus on interoperability between the business level and the shop floor level. The main purpose of interoperability between Enterprise Resource Planning (ERP) systems and Manufacturing Execution System (MES) is to improve the synchronisation between shop floor data and business information. ERP - MES interfacing speeds up the information flow between planning and process control systems. This allows tighter control of process timing and improvement of the quality of the production process. A priori detection and resolution of process disturbances into the schedule can reduce late deliveries. Better visibility of progress information leads to improved sales order control and follow up for the Customer Service department. Defining this interface requires an information analysis to sort out reliable and accurate process data for the ERP system and to define proper recipe information for the MES. Fig. 3 shows some examples of information that could be exchanged between the ERP level and the MES level.

[1] ISO/IEC 2382 (1993). Information technology -- Vocabulary -- Part 1: Fundamental terms

[2] IEEE STD 610.12 (1990). Standard Glossary of Software Engineering Terminology, IEEE, May, ISBN: 155937067X

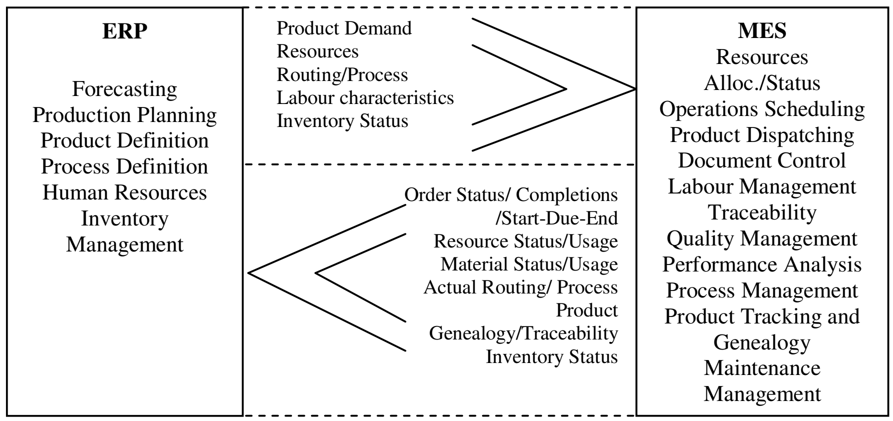


Fig. 3 Example of data that could be exchanged between an ERP and an MES

LinkForSap [6] is an example of interoperability solutions between ERP and MES systems. LinkForSap is a software package that enables information unification between the SAP R/3 ERP system and the Centum production control system. Even if the LinkForSap resolves the problem of interconnection in a computer integrated manufacturing system, this solution is valid for a specific ERP application, and can not be reused for another application. In the special case of the interoperability paradigm, point-to-point solutions that create single links between many applications are known to be expensive in term of maintenance; indeed the interoperability solution may become more expensive to maintain than the connected applications.

2.2 Workflows Interoperability Solutions

In manufacturing systems environment; in addition to ERP and MES, other kinds of applications could be involved. In the enterprise management level, workflows management systems, business process management systems can be used (Fig. 1). In this paragraph, some examples of interoperability solutions for this kind of applications are given.

The workflow management coalition (WfMC[3] [7]) defined an interface for the communication and the interoperation between workflow engines. A key objective of the WfMC is to define standards that will allow workflow management systems, henceforth called workflow product from different vendors to pass work item seamlessly between one another. The interface defined in [8] consists of the specification of all API calls needed to cover all cases and possibilities when two workflows can co-operate (e.g. exchanging status, synchronising execution, sending and answering queries). Wf-XML[9] is an interoperability standard defined by the WfMC working group, that combines the abstract commands defined by the interoperability interface defined by the WfMC, this standard consists in remote control workflow instances exchanging XML messages through http protocol. The

[3]www.wfmc.org

Wf-XML defines a set of request/response messages that are exchanged between an observer (which may or may not be a workflow management system) and a workflow management system (WfMS) that control the execution of a remote workflow instance. Several WfMSs incorporate the Wf-XML formalism to ensure interoperability of their engines (e.g. AFRICA [10]).

Shepherdson et Al. [11] proposes a cross organisational co-ordination of workflow and business process using software agents. In this proposal, a software agents is associated to each activity in a workflow or enterprise activity in a business process; this software agents is responsible of all communication that connect this activity to the others. A workflow provided by the software agent layer is called an agent enhanced workflow. This approach provides a mechanism to handle interoperability between workflows based on the interoperability between agents.

WfMC interoperability standards and most of existing techniques that enable business process or workflow interoperability are based on a message exchange paradigm (e.g. Wf-XML, BizTalk, FIPA ACL etc.). These solutions resolve only the particular case of syntactic interoperability (messages vocabulary, messages format, data types, etc.). To handle semantic concerns in workflow interconnection, Casati et al. ([12], [13]) propose a semantic specification of workflow interoperability.

Table 1 presents the classification of the studied solutions in terms of vertical or horizontal interoperability and also syntactic or semantic classification.

Table 1: recapitulation of the studied solutions for interoperability

Solution	Domain	Generic solution	Interoperability V/ H
LinkForSap[6]	ERP/MES interconnection	No	Vertical
Interoperability abstract specification [8]	Workflow interoperability	Yes (WfMC)	Horizontal
Wf-XML[9]	Workflow interconnection	Yes (WfMC)	-
Software Agents for Interoperability[11]	Workflow interoperability	Yes	Horizontal

In order to aid at taking into account the interoperability requirements during systems modelling phases, we introduce, in the next section, the concept of holon, in the context of manufacturing systems. This concept plays a central role in the specification of Holonic Process Models (HPM) as a specialisation of Business Process Models. We will then show how these models can be mapped into the Unified Enterprise Modelling Language (UEML [14], [15]) and the Business to Manufacturing Markup Language (B2MML) [16], which are modelling languages used to facilitate the interoperability of process models in the context of a Business to Manufacturing (B2M) approach.

3. The Holonic Approach and Manufacturing Applications Interoperability

In the manufacturing environment, numerousness of types of flows (data flows, physical flows, communication flows, event flows) introduces chaos in the organisation of flow traffic inside systems. In this context, specific problems appear, synchronising flows that handle related objects is one of those problems. In this section we define the concept of holon an its adaptation in manufacturing context as a first step towards the establishment of an approach for HPM. This approach aims to propose modelling principles for resolving information flows synchronisation in manufacturing systems, taking into account interoperability concerns during the modelling phase.

3.1 The Holon Concept in Manufacturing Process Modelling

The paradigm of "product + information" has been studied and defined as Holonic worldview [17]. The term "Holon" was applied to the manufacturing world, creating the Holonic Manufacturing Systems (HMS) community ([18], [19]). For this community a Holonic Manufacturing "is an autonomous and co-operative building block of a system for transforming, transporting, storing and/or validating information and physical objects. The Holon consists of an information processing part and often a physical processing part. A Holon can be part of another holon." ([20], [21]). A holon is then comprised of (Fig. 4):

- a physical part containing all the physical resources that constitute the holon.
- an informational part composed of all data that describe the holon, its knowledge and its behaviour.

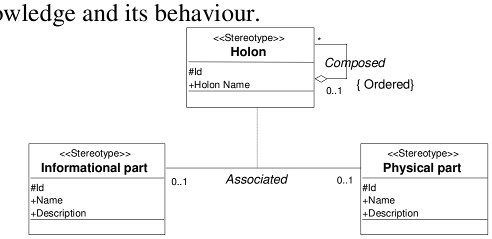


Fig. 4. The model of a Holon [22]

The previous class diagram shows a very high level abstraction of the concept of holon adapted to manufacturing systems. In a manufacturing environment, the concept of holon as defined above can be used to represent products that are equipped with information about their manufacturing, their life cycle and information relevant for its manufacturing scheduling and planning. Every processing states on the physical part of the holon is also stored in the informational part, that way the coherence between physical objects and their informational views is maintained along its life cycle. Moreover, integrating information dedicated to manufacturing decisions in the holon enables distributed decision and manufacturing systems flexibility. In the next section, we give a detailed definition for the holon concept in the manufacturing context.

The Holon concept adapted to manufacturing context. In [23], we have formalised the organisational view of a HPM putting forward holons flows and their interactions with processes. In this contribution, we focus on the structural view of holons.
We then consider that the information about a Holon is distinguished into two categories:

- Information describing the current state of the holon, this state contains three kinds of data handling attributes of space, attributes of form, and attributes of time.
- Information that concerns the holon but that does not fit into the state. This information is defined by a set of properties [19]; each property is defined by its name, description (use and unit of measure), and value.

Holons can be classified into two categories *(i)* simple holons and *(ii)* complex holons;

- Simple holons (elementary holons) are the combination of a single informational part and a single physical part.
- Complex holons (composite holons) are the result of the proceeding and treatment of one or more other holons, this proceeding can be a transformation of one holon to obtain a new one, or integrating a set of holons in order to compose a new one. Each composite holon can be defined as the output of the execution of a manufacturing process on one or more less complex holons.

In the example represented in Fig. 5, the holon *H1* is directly composed of the holons *H2* and *H3*, those holons are also composite holons. In Fig. 5, edges represent processes, nodes represent holons; composite are represented by continuous rings and elementary holons are represented by non-continuous rings.

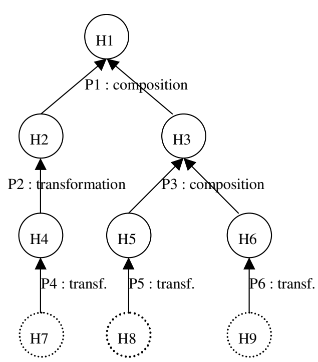


Fig. 5. A composite holon example

The specification and the definition of such holon are formalised by a class diagram showing how the concept of holon, together with its informational parts and physical parts, is linked to constructs coming from manufacturing systems domain, such as Process and Equipement (Fig. 6).

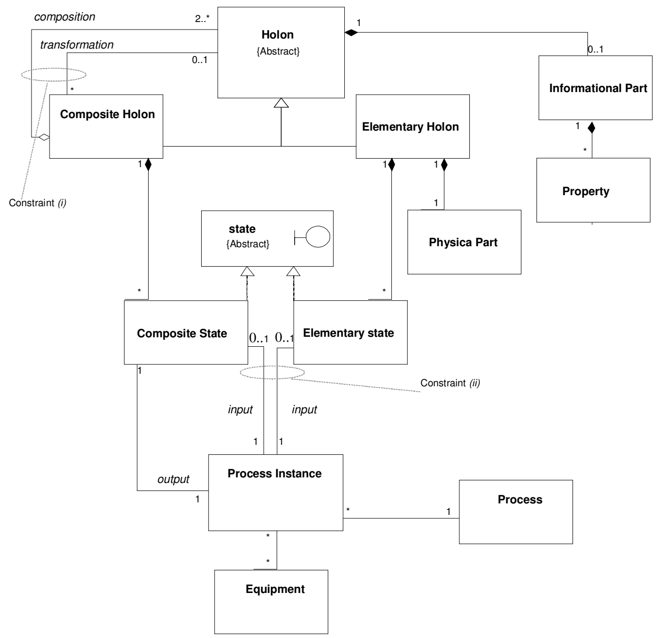


Fig. 6. Class diagram for the Holon model

Here is a brief description of the classes defined in the diagram represented in Fig. 6: The Class *Holon* defines basic attributes for both composite and elementary holons. An *Elementary Holon* is defined as a holon with no indication about his life cycle. For example a product, produced by external manufacturing systems does not give information about the processes needed for his manufacturing. In general, an elementary state is observed and associated to each elementary holon. A *Composite Holon* is a holon that has been processed through at least a single process during his manufacturing. Only, processes inside the domain of the enterprise are taken into account. A composite Holon can be obtained by two ways:

- Composing existing holons,
- Transforming an existing holon into one or more holons.

The UML interface *state* describes the current state of a Holon (composite or elementary). Every manipulation of a holon through a process (Process Instance) implies a change in the state of the holon processed. For the sake of traceability, it is possible to create a new state instance for each change in the existence of a holon, hence the hole life cycle of the holon can be stored. An *Elementary state* describes specific data that concerns the state of an *elementary holon*. The *Composite state* class describes specific data that concerns the state of a *Composite Holon*. A *Physical Part* is a reference to the physical part encapsulated in an *Elementary holon*. The *Process instance* refers to the execution of a process on a single holon, this class enables the description with high level detail of every processing of the holon(e.g.: elapsed time, start and end of the treatment, used equipment, needed personal). A *Process* describes an internal process, a process that is performed inside the studied domain. The *Equipment* class describes the equipment needed for performing a process instance. A *Property* contains all information concerning a *holon*, that can not be handled only using the *holon state.*

In the diagram represented Fig. 6, two constraints are expressed:
constraint (i) a composite holon cannot be at the same time a composition of other holons and the result of a transformation;
constraint (ii) a process instance input is a holon state, this state can be composite pr elementary but not both, hence a process instance cannot be associated to both a composite state and an elementary one.

Other constraints on class attributes can be expressed (example: *Precedence constraint;* the time of creation a composite holon cannot be anterior to the time of creation of the holons composing this holon.), but for the sake of simplicity those constraints where omitted from the diagram.

The Holon integration in a manufacturing process modelling tool. To apply the Holonic model defined above in real use cases, an implementation has been proposed. This implementation consists on integrating this model in an enterprise process modelling tool named MEGA Suite[4], which is a business process analysis, modelling and development environment.
MEGA Suite is based on its own meta-model which can be specialised for specific needs. We used that facility to embed our own Holon model into that environment in order to test the usability of our proposal on specific application test cases. This implementation enables using Holon flows exchange when modelling processes, specifying then a Holonic Process Model.

In the next section, we show how the holon concept and the model defined can match with other reference models and meta-models of languages used in the area of enterprise integration and enterprise modelling. This matching between the different concepts enables mappings at model level, thus models using holons could be translated and integrated into models based on other standards.

3.2 Integration and Interoperability with other Models

The main objective of this section is to show that models based on the holon concept defined in section 3.1 could be expressed and transformed into models based on existing data exchange standards and other unified languages. This transformation will enable model and data exchange between applications that are based on holon models and different enterprise applications from different levels in the enterprise. To experiment the holon integration and interoperability with other enterprise modelling frameworks, two standards have been chosen:

[4] MEGA Suite, MEGA International, www.mega.com

- The UEML a Unified Enterprise Modelling Language used at the organisational level of the enterprise;
- The B2MML language is an implementation of the IEC/ISO 62264 standard [24] used for interfacing the manufacturing control and execution systems with higher level systems.

Mapping Holon with the Unified Enterprise Modelling Language. The Unified Enterprise Modelling Language (UEML) is the result of the UEML project [25] which aimed to solve the problem of numerousness of enterprise modelling (EM) languages [26]. The project defined a Unified Enterprise Modelling Language as an Interlingua between EM tools. The meta-model of UEML1.0 ([14], [15]) defines the set of most relevant concepts and notions for Enterprise modelling. The UEML meta-model defines a class *Object*, with two sub-types:
- *Information Object*: describes object that contain only data. Information objects can be organised in "control flows" (constraint flows or trigger flows);
- *Resource*: can be divided into two sub-types *Material resource* and *Human resource*. Resources are organised in *Resource flows*.

A trivial mapping consists in representing *Informational part* (resp. *Physical part*) using the concept *Information Object* (resp. *Material resource*), to represent holons we need a specific sub-type of the class *Object*, this specific concept must be the link between the *Material resource* (*Physical part*) and the *Information Object* (*Informational Part*).

In the UEML meta-model, an activity represents a generic description of a part of enterprise behaviour that produces outputs from a set of inputs. It represents a set of similar activities executions. The concept of “Activity” defined by the UEML constructs can be used to express the notion of “Process” which is defined in the holon model. Table 2 summarise the mapping between concepts defined in the holon model into UEML constructs.

Table 2: Correspondence between Holons related concepts and UEML concepts.

Holonic Process Model Meta-model	**UEML Meta-Model**
Holon	Object
Informational Part	Information Object
Physical Object	Material resource
Process	Activity

Mapping with the B2MML language and the IEC/ISO62264 standard: Business To Manufacturing Markup Language (B2MML) is an XML implementation of the IEC/ISO62264, it consists in a set of XML schemas written using the World Wide Web Consortium's XML Schema language (XSD) that implement the data models defined by the IEC/ISO62264 standard. The IEC/ISO 62264 is the standard specifying the exchange of data and models interfacing the shop floor level into the enterprise. It is composed by six different parts designed for defining the models and interfaces between enterprise activities and control activities. Each model concerns a particular view of the integration problem. Those models show increasing detail level in the manufacturing system. They can be classified in to two categories; operational models or resource models. To map the holon concept with the IEC/ISO 62264 models, two views are possible:

- Genealogical view: this first view concerns the genealogy of the holon, which describes relationships between holons (composition and transformation). Information handled in this view corresponds to the "Product Definition Information", as defined in the standard;
- Process view: allows representation of the lifecycle of the holon and the different processes involved during each step in the manufacturing cycle. This view fits into the "*Product production rules*" concept of the IEC/ISO 62264 standard, especially the "*Product segment*" concept, which can manage information about assembly steps and assembly actions for discrete manufacturing.

Both the genealogical and the process view are relevant for managing holons traceability along its lifecycle. In the IEC/ISO 62264, the material model is a resource model that defines the actual materials, material definitions and information about classes of material. Material information includes the inventory of raw, finished, intermediate material and material decomposition into lots and sublots. The IEC/ISO 62264 material model is the most adequate to express and handle the information about genealogy of holons (products), thus it should be used to represent the genealogical view of holon. In the IEC/ISO 62264, the notion of product segment defines the values needed to quantify a segment for a specific product, such as the number with specific qualifications. In the holon model, the class "*Process instance*" groups all information that concern the manufacturing of a specific holon. Since we consider that a holon represents a product, we can then assume that the "*Process instance*" of a specific holon describes the product segment of that holon (product). This means that all concepts associated to the *Process instance* class, can match the concepts associated to Product segment.

Table 3 shows the mapping between the genealogical aspect and the process view of the Holon model into the material model of IEC/ISO 62264 standard.

Table 3: Correspondence between Holons related concepts and IEC/ISO 62264 concepts

	Holonic Process Model	**IEC/ISO 62264**
Material Model mapping	Holon	Material sublot
	Holon Flow	Material lot
	Informational Part	Material definition
	Properties	Material lot property and Material lot property definition
Product Definition Model mapping	Process Instance	Product segment
	Equipment	Equipment specification

Mappings in action for interoperability. The previous paragraph defines mappings between the holonic model with UEML and IEC/ISO62264 models. In order to validate these concepts, we are currently developing an automatic translator based on proposed mappings from our holonic model into the other models. To describe objects and models manipulated by each meta-model the XMI [27] (XML Metadata Interchange) language has been chosen. XMI is a widely used interchange format for sharing objects using XML [28]. The XMI formalism proposes a solution enabling meta-models description in XML syntax. Moreover XMI enables also description of objects and their instances based on those models.

The XMI interface enables generating different XML files with a specific structure for each one using adequate translators, these structures are defined by the adequate standards, as shown in Fig. 7. The B2MML and the UEML XML schemas are used to generate valid and reusable documents from holonic process diagrams created in the managed in the MEGA Case tool, those documents can the be imported in applications compatible with UEML or B2MML standards.

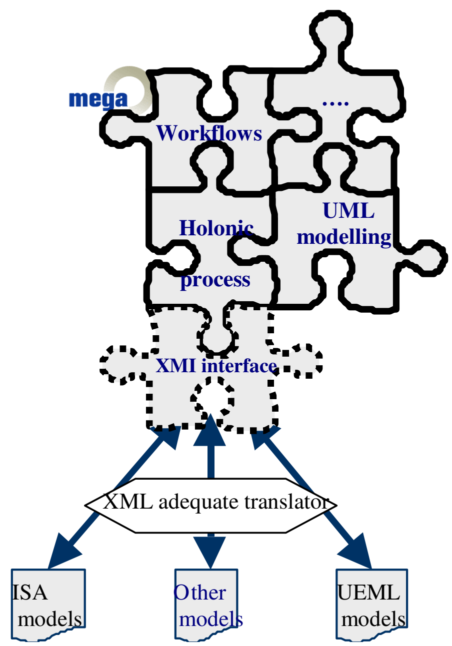


Fig. 7. Transformation of models

4 Conclusion

In this paper we defined an approach for specifying Holonic Process Models. In these models, the holon represents the key notion. This concept enables maintaining coherence between the physical objects and their informational views, and thus between informational flows and physical flows. This approach aims also at increasing model abstraction in order to simplify its initial comprehensiveness and ensure coherence between different views of manipulated objects. We gave a proposal to map and integrate the holon concept in existing enterprise modelling frameworks for interoperability concerns. In this proposal, we focus on ERP interconnection with an MES in order to exchange manufacturing and production related information. In order to validate our approach, further work will be to develop translators from our XMI interface to the respective B2MML and UEML XML files. Perspectives for this work concern the interconnection with a WfMS, in order to exchange business process modelling information and then enabling a process oriented execution of enterprise models.

References

[1] Molina A., Chen D., Panetto H., Vernadat F. and Whitman L. (2004). Enterprise Integration and Networking: Issues, trends and vision. IFIP International Conference on Enterprise Integration and Modeling Technology (ICEIMT'04), Kluwer Academics Publisher, October 9th-11th, Toronto, Canada.
[2] Petrie C. *Entreprise Modeling and Integration.* C. Petrie (ed). McGraw-Hill. 1992.
[3] AMICE, 1993 AMICE (eds.). CIMOSA: Open System Architecture for CIM, 2nd revised and extended edition. Springer-Verlag, Berlin, 1993.
[4] Formal Support for Representing and Automating Semantic Interoperability Yannis Kalfoglou and Marco Schorlemmer.
[5] Vallecillo, A., Hernàndez, J., and Troya, J. M., Component Interoperability.
[6] Yutaka S. Information unification between enterprise resource planning system and production control system, Yokogawa Technical Report, English Edition, N° 25 (1998)
[7] Workflow Management Coalition: The Workflow Reference Model. Document Number TC-1003, January - 1995
[8] Workflow Management Coalition: Workflow standard – Interoperability abstract specification. Document Number WFMC-TC-1012, 26-October-1995
[9] Workflow Management Coalition: Workflow standard – Interoperability Wf-XML Binding. Document Number WFMC-TC-1023, 1-May-2000
[10] Zur Muehlen, Michael; Klein, Florian: AFRICA: Workflow Interoperability based on XML-messages. CAiSE 2000 Workshop on Infrastructures for Dynamic Business-to-Business Service Outsourcing (ISDO '00). Stockholm, Sweden, June 5-6, 2000
[11] Shepherdson J.W., Thompson S.G. and Odgers B. (1999). Cross Organisational Workflow Co-ordinated by Software Agents Work Activity Co-ordination and Collaboration (WACC '99) Workshop Paper, February 1999
[12] Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G. Semantic workflow interoperability. In Proc. of the 5th Conf. on Extending Database Technology (EDBT), pages 443--462, 1996.
[13] Casati, F.; Ceri, S.; Pernici, B.; Pozzi, G.: Deriving active rules for workflow enactment. In: 7th Intl. Conf. on Database and Expert Systems Application. 1996, pp. 94-115
[14] Berio G, et al. D3.2: Core constructs, architecture and development strategy, UEML TN IST – 2001 – 34229, March 2003
[15] Panetto H., Berio G., Benali K., Boudjlida N. and Petit M. (2004). A Unified Enterprise Modelling Language for enhanced interoperability of Enterprise Models. Proceedings of IFAC INCOM Symposium, April 7th-9th, Salvador de Bahia, Brazil
[16] The World Batch Forum. Business To Manufacturing Markup Language (B2MML), version 02, 2003.
[17] Koestler A. (1967) The Ghost in the Machine Arkana, London.
[18] Marik, V and Pechoucek, M (Eds.) (2001). Special issue on industrial applications of Holonic and Multi-Agent Systems. *Journal of Applied System Sciences*. **2/1**.
[19] Morel G., Panetto H., Zaremba M.B. and Mayer F. (2003). Manufacturing Enterprise Control and Management System Engineering: paradigms and open issues. IFAC Annual Reviews in Control. 27/2, 199-209, December
[20] Wyns, 1999, Reference architecture for holonic manufacturing system, Ph.D. Thesis, Katholieke Universiteit Leuven

[21] Seidel D. and Mey M. (1994). IMS – Holonic Manufacturing Systems: systems components of autonomous models and their distributed control.

[22] Gouyon D., Simão J. M., Alkassem K. and Morel G., (2004). Work in progress for product driven manufacturing automation, Proceedings of IFAC INCOM Symposium, April 7th-9th, Salvador de Bahia, Brazil

[23] Baïna.S, Panetto H. and Morel G. A holonic approach for application interopearbility in manufacturing systems environment, *In Proc of the 16th IFAC World Congress, Prague, July 4-8, 2005.*

[24] ISO/IEC 62264 (2002). ISO/IEC 62264-1:2002. *Enterprise-control system integration, Part 1. Models and terminology*, ISO

[25] UEML. (2003). Unified Enterprise Modelling Language (UEML) Thematic Network. IST-2001-34229, www.ueml.org.

[26] Vernadat F.B. (1996). *Enterprise modelling and integration: principles and applications.* Chapman & Hall.

[27] Object Management Group: XML Metadata Interchange Specification, version 2.0, May 2002.

[28] W3C. World Wide Web Consortium, Extensible Markup Language (XML) 1.0, 1998

Knowledge map for distributed information system re-engineering

Jorge Pardellas

University of Geneva, CUI, 24 rue General Dufour,
CH-1211 Geneva 4, Switzerland
jorge.pardellas@cui.unige.ch

Abstract. The distributed information system design decision has always been taken in the implementation step. The contribution of designer for this decision can decrease the evolutionary and the corrective maintenance. This designer's contribution can be realised through the specification of integrity constraints at distribution step. In this paper, we illustrate the contribution of such an approach for the design of distributed information system and we explain how this approach will be taken into account for the evolutionary re-engineering of such systems. This approach relies on the knowledge maps established between designer and developer that encapsulate the set of information and thus contribute to the distribution implementation.

1 Introduction

Increasingly today, organisations and their environment, due to their dynamic behaviour, generate new informational and structural needs that should be satisfied for their survival. The unification of organisations and the effect of the globalisation imply an integration of multiple data coming from various structures to coordinate them and thus ensure their integrity. This integration is done through their different correspondent information systems. The information systems have a major role intra and inter organisational by ensuring the collection, storage, sharing and the integrity of the data. In addition, an operational and robust information system must provide the right information to the right person at the good moment and with the lowest cost [7]. Information system has to be conceived as the principal actor in the decision-making process of any organisation [5].

These information systems, therefore, must imperatively be able to evolve with the structural and informational dynamism in order to satisfy the updated organisational and technical requirements. Because of their conceptual nature, the traditional information systems, generally centralized, are not easily maintainable and consequently cannot be adapted to these changes in order to reflect adequately the new requirements [3]. Hence, the traditional information systems do not support the evolution of the organisation's conceptual diagram, i.e. the internal data and consequently the internal database diagram, the integrity constraints and the specifications of the dynamic aspects. To contribute to the maintenance of coherence in the database, it is necessary to specify constraints, which the database must respect to preserve the data integrity. These constraints change and should be changed in order to meet the new needs. In fact, the evolution of the organisational environment or/and the informational environment often implies a radical change in the thinking of these constraints, which control these environments.

The maturation of database management system (DBMS) technology has coincided with significant developments in distributed computing and parallel processing technologies. The result is the emergence of distributed information systems based on distributed database management systems and distributed applications system. A distributed information system is a collection of data, transactions, integrity constraints and sites [2]. Data is concretized by a set of multiple, logically interrelated databases distributed over a computer network. This set is managed by a distributed database management system, which makes the distribution transparent to the users. Some researches on integrity constraint checking in distributed database environments addresses the problem of checking the global integrity constraints locally. [1] proposed the demarcation protocol which relies on storing some extra information about data on remote sites and the solution applies only to a limited class of integrity constraint. [2] presents ways to produce sufficient integrity constraint verification tests given a modification to the database and the entire original database. [6] describes techniques for distributing global constraints between sites in a way that reduces communication time.

Constraints of distributed information system are distributed on various levels. They can be implemented on the different existing levels of the system. In architecture such n-tiers levels, constraints can be implemented in graphical interfaces, application level or the data level. It is important to make the evolution of these integrity constraints easier to ensure the survival of information system. The information system evolution based on its integrity constraints evolution is difficult to manage because this evolution is complex and still controlled neither by the distributed database management system nor by the design and development tools of information system. The control of this evolution is required since it is the guarantor of the data's quality and the information system's performances [7].

The specification of integrity constraints in a distributed information system can cause problems at three different levels: transaction design, development and maintenance of integrity constraints. Firstly, the transaction design step guarantees the coherence of each transaction with respect to the integrity constraints. It has to meet the performance requirements ensuring a minimal data flow during execution of an integrity constraint using shared data between different sites. Secondly, the development of integrity constraints has to respect the design step and has to be executed in order to ease the future maintenance and the reliability of the constraints. Finally, the general problem is to check that the integrity constraints based on distributed data among different sites, are verified after any performed transaction.

2 Distribution framework

In order to answer the distribution problems, we needed to develop a framework and an associated suite of methods that enable designers to have assistance for the database schema specification while considering the different parameters involved.

2.1 Overlap knowledge pattern

We define an overlap knowledge zone between designers and developers in which they share common information and more important common distribution decisions. The distributed information system design knowledge pattern treats the knowledge surrounding the design step. This knowledge represents information obtained from the analysis step and expresses the main directions of the system. The distributed information system implementation knowledge pattern contains different generic knowledge about the constraints imposed by the different technical platforms. It specifies also the different performance and quality of service objectives. The distributed information system overlap knowledge pattern is essentially composed of shared decisions that must be taken both by designers and developers. It contains information about the integrity constraint design and the way of implementation, data locations and the different site relations. The designer's knowledge can contribute to the specification of the distributed information system through the different information extracted from the analysis at design step, i.e. triggering site, end-user queries, design justification, critical data, constraint weighting, validation site, etc. The developer's knowledge, i.e. the technical constraints he faces, has to be taken into consideration as well. This knowledge from both actors has to be shared in order to design and implement a distributed information system and its integrity constraints. Considering this common knowledge will help building evolutionary and easily maintainable systems.

2.2 Distribution algorithms

Several algorithms are proposed FKDO: Foreign Key Dependencies Optimisation, ICDO: Integrity Constraint Dependencies Optimisation, GIDO: Global Integrity Dependencies Optimisation are the main component of the Overlap Knowledge Pattern. The first algorithm is based on the foreign key distribution optimisation. Its aim is to decrease the number of the different site relations resulting from foreign key matching. The second algorithm treats the other different integrity constraints. Its result is a set of distribution solutions. The different generated solutions minimise the number of site connections and thus their dependencies. They also reduce the data flow and deliver better system performance and better answering time. Finally these algorithms structured in a global framework, that takes in consideration the various constraints' importance, recalculate the different solutions while matching every constraint with a weighting value. This value evolves continuously according to the frequency of the constraint call. Our different algorithms are the principal component of our purposed methodology for the design of distributed information system based on integrity constraints dependencies optimisation. The first goal of this approach is to give the designer a real view of the distribution system. This is done by generating the different distribution sets and comparing them according to their respective transaction costs. The second goal is to underline the most advantageous solution while giving the possibility to the designer to interact and influence this solution. This flexibility is concretized by the ability of the designer to fix needed data on predefined site and to order the different integrity constraints by determining the respective weighting. The third goal is to obtain the lowest distribution cost while maximizing the system integrity and availability.

3 Future directions

From the different knowledge patterns, the focus has been brought to the overlap knowledge pattern by defining different algorithms to allow the implementation of the distributed database diagrams while relying on the various components of the pattern. We will now focus on the other different components of the design and implementation knowledge pattern. We need to specify the implementation knowledge pattern to be able to re-engineer the conceptual diagram through the use of the integrity constraints. Indeed, the proposed distribution approach weight the transaction cost in an arbitrary way. During runtime, the system's data flow created by an integrity constraint could lead to rethink the conceptual diagram. By this way, the weighting criteria will be adjusted according to the actual call of every constraints and the regulation of the distribution combination will satisfy the reel system requirements.

The development of a CASE tool integrating the different distribution approaches is a very important issue. The FKDO algorithm is now supported by our implemented tool M7Tool. The extension of this tool to ICDO and GIDO need to be done to provide a complete tool to assist designers in the specification step of a distributed information system.

References

1. Barbara, D., Garcia-Molina, H.: The Demarcation Protocol: A Technique for Maintaining Arithmetic Constraints in Distributed Database Systems. In Extending Database Technology Conference, LNCS 580, pages 373--397, Vienna, March, 1992.
2. Bry, F., Manthey, R., Martens, B.: Integrity Verification in Knowledge Bases. In Logic Programming, LNAI 592 (subseries of LNCS), pages 114--139, 1992.

3. De Weger, K. M., A. Vissers, C.:Issues in design methodologies for distributed information systems, Center for Telematics and Information Technology, University of Twente, The Netherlands,1999.
4. Leitzelman, M., Dou, H.: Essai de typologie des Systèmes d'Information,. International Journal of Information Sciences for Decision Making N°2, April 1998.
5. Leonard, M: Database design theory, Macmilan Publication Ltd, Macmilan Computer science Series, 1992.
6. Qian, X:. Distributed design of integrity constraints. In Proceedings of the Second International Conference on Expert Database Systems, pages 205-226. The Benjamin/Cummings Publishing Company, 1988.
7. Snene, M.: Knowledge Patterns of Distributed Information Systems – The Case of Distribution Design and Implementation Based on Integrity Constraints Optimisation, PhD Thesis n°578, Computer Science University Center, Geneva, 2004.

The ISDA model for integrating the Static and Dynamic Aspects of Information Systems

Thanh Thoa Pham Thi

University of Geneva, CUI, 24 rue General Dufour,
CH-1211 Geneva 4, Switzerland
thanh.phamthi@cui.unige.ch

1 Context

Current methods for information systems modelling often focus on the different aspects of information systems (IS). The methods, for example, Merise [10], IDA[3] and UML[4] are interested in the Static, Dynamic, Organizational, and Architecture aspects. Each aspect may dispose one or several models, which have different formalism and interpretation.

This paper concentrates on the Static and Dynamic aspects, which are the most fundamental aspects at the Informational level. At this level, the Class model represents the Static aspect; meanwhile the Dynamic aspect is represented by the State-chart or Object life cycle model.

These two types of models are indeed deeply interrelated. However, due to the differences between the two models, the verification of consistency of these models becomes an interesting challenge, especially in the case of the evolution of the models.

In the literature, many works have concerned with this challenge that can be grouped into two categories: verifying the consistency in each model [3][5] and verifying the consistency between different models [8][9]. Most of them used formal methods and transformed the specifications from source models, which have different formalisms, into a common formalism. In general, the Z formal language and First-order logic are used as common formalisms. When the specifications are transformed into the common formalism, they are verified the consistency within /between the two aspects. The method of [8] is different. The common formalism is a graph consisting of nodes and edges with multiplicities, and then a system of linear inequalities is derived from the graph for solving the consistency verification.

In our standpoint, the verification task can be avoided if we take into account this challenge at the modelling phase. It is the reason why we propose in this paper a unique model: **the ISDA model** that can be used to specify the Static aspect, Dynamic aspect as well as interrelations between them. Furthermore, the formalism of this model must not be more complicated than the formalisms of other methods so that IS professionals from different background can easily work with it. Besides, the proposed model should be easily implemented.

This model can be applied in the recently research and industrial domains such as the interoperability of information systems, components for enterprise systems, and evolution of information systems.

2 The ISDA model

As mentioned above, the main purpose of our work is to propose such a unique model for representing both the Static and Dynamic aspects of IS. This model is called the *Integrating Static and Dynamic Aspects model* (in short the ISDA model).
The concepts in our model are inspired from the object-oriented paradigm with some improvements to support more integration and evolution. In fact, this makes the differences between our model with other methods and therefore expresses the advantages of our approach.

This section presents the outline concepts of the ISDA model. Firstly, we present concepts of the Static aspect. Secondly, we continue with the concepts of the Dynamic aspect. Finally, we show how to present interrelations between the two aspects in our model.

2.1 Static aspect

In our approach, the *Class model*, which represents the Static aspect of IS, describes the structure of classes and relationships between them.
There is only one relationship type in the Class model in our approach: the ***existential dependency*** relationship, it helps supporting the evolution of the specification [6][7].
There is an existential dependency relationship from the C1 class to the C2 class if, for every object in the C1 class, the existence of this object depends on the existence of one and only one object in the C2 class. This dependency is permanent in the life of objects in the C1 class.

On the other hand, the well-known concept: *Specialisation* or *Is-A* in current modelling methods have often implemented by the inheritance mechanism. Consequently, once an object is specified in a class, it stays permanent in this class.
In our approach, we propose the *Dynamic specialisation* (DS), which is a particular case of existential dependency. The main objective of the DS is to authorize an object to change the class, to which it belonged to, in its life cycle. Consequently, the dynamic specialisation is more flexible than the specialisation concepts in other approaches:

- Objects can move between classes in the Dynamic Specialisation graph (DS graph). A DS graph is defined as a set of classes linking by dynamic specialisations;
- When objects move from a class (called the Super-class) to a sub-class of this class (called the Sub-class), it may be *active* or *inactive* in the Super-

class. When an object is inactive then one cannot execute any method on it, but one can consult it;

- As mentioned above, an object may stay in several classes in a DS graph. So it depends on the fact that in which class, one desires to access the object then the object shows/ hides its different informations. The concept ***Access-View*** is defined to describe this situation. An Access-View of a class consists of all properties defined on the C class and *certain* informations (attributes/ methods) of super-classes of C. There is no inheritance in the dynamic specialization [2].
- DS is not a reflexive relationship, i.e. if C1 is a super-class of a C2 class then the C2 class cannot be a super-class of the C1 class. In some situations of DS, an object may move from a sub-class to a super-class; it requires keeping the trace of the object. The concept ***Loop*** is defined for specifying this situation.

Concerning the formalism, the Class model in our approach is probably simpler than other approaches, because it uses a minimum set of concepts. Thanks to DS, the ISDA model can express more remarkable situations in which objects can change its class, can hold its traces, and can return to be an object of its super-class.

2.2 Dynamic aspect

The *Object Life Cycle* (OLC) represents the Dynamic aspect in the ISDA model. An OLC is presented by a bipartite graph, which is called *Node-Star Net* (NSN). A node corresponds to a *state*, and a star corresponds to a *transaction.* A transaction changes states of objects; it may be a function, a processus of systems or even a method of class.

In fact, there are some differences compared with OLC or state-chart in other methods [4][11]:

- We can indicate that an object holds former states when it changes from one state to other state. This primitive is not supported by other methods;
- Designer can define pre-conditions and post-conditions on transactions, meanwhile in the Petri net, there are only the conventions such as AND, XOR on logic connectors.

Concerning the formalism, the Node-Star Net is as complex as the Petri net, however it is able to express more situations above.

2.3 Integration of two aspects

We specify the correspondences of an object life cycle of a class to a DS graph with the respect to the order of states' change. Each state of an object life cycle of a *C* class corresponds to a class in a DS graph, which has *C* as its root class. If *S1* is a source state and *S2* is a destination state of S1 transformed by the transaction *T*, then the *C1* class (corresponding to *S1*) is a super-class of *C2* (corresponding to S2).

By this way, the dynamic specialisation concepts is used to present interrelations between the Static aspect and the Dynamic aspect. So this concept is a corner-stone of the integration of the Static and Dynamic aspects.

3 Conclusion and future works

We have presented the concepts of the ISDA model that aims at supporting the consistency between the Static and Dynamic aspects of information systems at the Informational level. This model uses the dynamic specialization to express the interrelations between these two aspects.

We also consider that the Organizational aspect is a very important aspect of information systems. Therefore, in the future works, we will take into account the consistency of this aspect with two Static and Dynamic aspects.

At the time being, we are developing an environment that allows specifying the specifications of the ISDA model and then implementing them automatically. Accordingly, we are applying the ISDA model in several application domains such as the interoperability of information systems, components for enterprise systems, and workflow applications.

References

1. Al-Jadir L., Evolution-Oriented Database Systems, Thèse Faculté de SES Université de Genève, 1997.
2. Al-Jadir L., Léonard M., If we refuse the Inheritance…, Proc. Int. Conf. on DEXA, Firenze, 1999.
3. Bodart F. et Pigneur Y., Conception assistée des systèmes d'information – Méthode - Modèles- Outils, Masson Paris 1989.
4. G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley 2000..
5. Formica A., Frank H., Consistency of the static and dynamic components of object-oriented specifications, Data & Knowledge Engineering, Volume 40, Issue 2, February 2002.
6. Léonard M, M7 : une approche évolutive des Systèmes d'Information, Inforsid à La Garde, juin 1999.
7. Léonard M, IS engineering getting out of classical system engineering, ICEIS, Angers France 2003.
8. Pascal Fradet, Daniel Le Métayer and Michaël Périn, Consistency checking for multiple view software architecture, ESEC/ FSE'99 volume 1687 of Lecture Notes in Computer Science, ACM Press.
9. Holger Rasch, Heike Wehrheim, Consistency between UML classes and Associated State Machines, workshop "Consistency problem in UML based software development", the 5th IC on The Unified modeling language, the language and its application, October, Dresden, Germany, 2002.
10. Hubert Tardieu, Dominique Nancy et Daniel Pascot. Conception d'un système d'information : construction de la base de données. Editions d'Organisation, 1979.
11. Nanci Dominique, Espinasse Bernard, Ingénierie des systèmes d'information : Merise, Deuxième génération, Vuibert Paris 2001.

Process oriented approach for enterprise information system design
Abstract of PhD work

Jihed TOUZI

Centre de Génie Industriel
Ecole des Mines d'Albi Carmaux
Campus Jarlard Route de Teillet
81 013 Albi Cedex 09-France
jihed.touzi@enstimac.fr

1 Introduction

The information system (IS) of an enterprise(or collaborative enterprise network), constitutes an essential support of its functional processes. For this reason, this platform of management and pooling of enterprise data, presents a character which is extremely specific to each enterprise(or collaborative enterprise network). The definition of the "good information system" for a given enterprise constitutes a difficult task.

Our objective in this PhD work, is to define a “translator” that make possible, starting from a certain number of models, related or not to the enterprise (process models, enterprise models, generic models of information system, ect.) and using enterprise knowledge, business rules, company culture, current information system modules to emerge the most precise and the most usable specification of an information system(model) which is relevant and specific to the enterprise.

Our research work is centered on the “information system design” field. It’s based on three axes. The first axis is about **Formalisms**, we mean IS architecture, IS generic model, enterprise modeling, process modeling and exploitable process models. The second axis concerns **Knowledge**, we mean ontology and structured knowledge bases. The third axis is based on **Integration Tools**, we mean concepts, methods and software for integration and interoperability.

2 Research axes and IS design approach

In the first axis we started exploring enterprise modeling methods/methodologies. An *enterprise model* is a computational representation of the structure, activities, processes, information, resources, people, behavior, goals, and constraints of a business. So before we have to think about which kind of information system is good for one enterprise ,we must know “every thing” about it. Methods/methodologies like CIMOSA,GRAI and OLYMPIOS , give a complete representation of the structure of the enterprise. a process is represented by a formal sequence of procedures, activities and steps. However, process models in the functional view, are not expressive enough to reflect dynamism, flexibility and the need for interoperability of enterprise process. Some process modeling languages like BPMN (*Business Process Management Notation*) which is on the way to become a standard in process modeling , have a rich vocabulary. Moreover BPMN offer integration capability with enterprise data and applications. In this context we aim to find one process formalism that permit describing perfectly enterprise process and integrating completely other enterprise views (information, organization and resources). Formal exploitation of process models remains a delicate point.

In the second axis we admit that (enterprise) ontology present key aspects of an enterprise. Ontology contains business knowledge, resource structures and person capabilities. This ontology (enterprise) is semiformal; it provides a glossary of terms expressed in a restricted and structured form of natural language supplemented with a few formal axioms. We started by study ontology build phases and existing ontology types. We are interested by TOVE(Toronto Virtual Enterprise) project. It describes an object enterprise ontology based on structure, behavior, authority, empowerment and commitment competencies.

In the third axis, we look for workflow enactment of the resulted information system model. For this, we must integrate different enterprise modules (or components). Enterprise modules are implemented building blocks or systems (products, or families of products), that can be utilized as common resources in enterprise engineering and enterprise integration. As physical entities (systems, subsystems, software, hardware, available human resources/professions) such modules are accessible in the enterprise, or can be made easily available from the market place. EAI (Enterprise Application Integration) concepts are a solution for integrating heterogeneous applications using data exchange language like XML (eXtensible Markup Language).

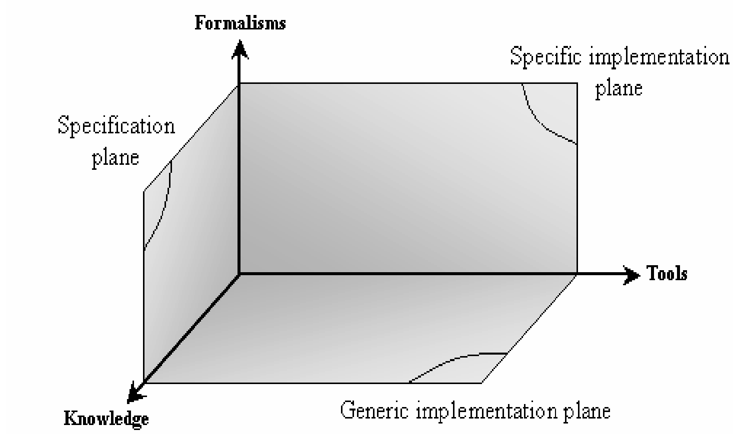


Fig. 1. Process research axes concerning the information system translation. The F*ormalisms* and *Knowledge* axes deal with *specification* issue while *Tools* axis brings *implementation* notion to others axes. Indeed, *Formalisms* and *Tools* are linked by specific process or IS or enterprise modeling (or implementation). Similarly, *Knowledge* and *Tools* are connected by generic concepts of implementation.

The **process-based approach** is a strong feature of our current research, but formal exploitation of process models remains a delicate point. This approach aims to translating a confident process model into a pertinent IS design model. This translation is based on "general knowledge", which includes *generic IS models*, *EAI components* and *models of existing IS*. The components of that "general knowledge" should be used by the semi-automatic translator during the construction of the pertinent IS model. They are guiding the translation by providing additional (and essential) elements for the IS model design.

Technically, if BPMN seems to be the best candidate for pertinent workflow modeling, it is difficult to find trustable ways of exploiting generated models. XML is certainly an adequate medium for data spread but obtaining BPEL4WS files from BPMN models is not a trivial action (BPEL4WS is Business Process Execution Language for Web Services and is a XML language designed for business process). Thus, if we want to use XML files to generate (using the translator) UML IS models (in XMI), we must be able to obtain BPEL4WS models (or such XML business models).

3 Conclusion

the IS model generation cannot simply be a compilation of workflow models because the single process model does not contain the whole information required for IS model design. Indeed, the objective is not to translate a process expressed in language A into a process expressed in language B. It is to create an IS model (thus located on a different level) from specific data contained in process models, by using general knowledge surrounded by *reference models*, *existing IS models* and by every other kind of knowledge that could be useful including information provided by actors to the translator.

References

1. Morley, C.: La modélisation des processus : typologie et proposition utilisant UML. In Processus et Systèmes d'information – Journées ADELI, Paris, France (2002).

2. Darras, F.: Proposal of a formal framework for the design and use of an Enterprise Resource Planning software. PhD thesis – INP Toulouse, (2004).

3. Vernadat, F.: Enterprise modeling and integration. Chapman & Hall (1996)

4. Vernadat, F. UEML: Towards a Unified Enterprise Modeling Language. In Conception, Analyse et gestion des Systèmes Industriels, pages 3–10. 3ème Conférence Francophone de MOdélisation et SIMulation - MOSIM'01, (2001)

Towards DRM/DPM enabled interoperable Enterprise Information Systems

Michel Pawlak,
Michel.Pawlak@cui.unige.ch

University of Geneva
Centre Universitaire d'Informatique
Geneva, Switzerland

Abstract. Modern enterprise information systems are heterogeneous environments combining resources coming from different sources. Multiple domains are involved, and as such the ability of information systems to be interoperable, the ability to ensure persistant security as well as the ability to reflect the dynamics of the corporate environment are mandatory. The definition and management of static access rights through access control lists dispatched accross multiple and evolving environments is difficult and not sufficient anymore. Further, information systems are now expected to ensure regulatory compliance. In this context, security of information systems has to be extended and adapted to provide an interoperable global security environment, reflecting strategical security needs and applying them at the operational level.
This paper discusses how Digital Rights and Policy Management can enable interoperable and dynamic policy driven security and presents the benefits of such a global interoperable security scheme over traditional security schemes.

1 Introduction

Nowadays, modern enterprise information systems are an interconnection of heterogeneous systems like frameworks, knowledge management systems, enterprise resource planing, databases, datawarehouses, etc. Resources can be distributed over multiple sites belonging to different companies. Different technologies and approaches are used to access these resources, to communicate and provide service like web services or mobile agent systems. Involved processes have to cross these heterogeneous systems and be able to to adapt to the different security schemes and authentification systems that may be combined. Thus resource security and security interoperability are extremely important issues.

Further, currently, increasing emphasis is put on regulatory compliance. For instance Sarbanes-Oxley Act (SOX [1]) auditors can demand to see auditing records of virtually all computer access. This increases the need for adapted security solutions to control access to enterprise sensitive ressources. SOX is a first glance of how economical and legal environments will influence tomorrows enterprise information systems.

Future information systems will have to be able to reflect strategical issues like corporate governance, compliance, liability, policies, etc. Information systems will have to be able to capture and apply these issues at an operational level, in order to ensure valid system execution. Moreover, to constantly reflect reality, future information systems will have to be able to evolve dynamicaly with the environment. In this context, issues concerning persistent protection of ressources, adaptable policy driven control of distributed ressources and interoperability are not disjoin domains. Traditional security approaches through static control access lists are not sufficient anymore.

Nevertheless, interoperability does not only concern IT and IS related issues. Aligning the business and enterprise strategy with IT and IS issues is where future interoperability issues reside. Flexibility, adaptability, persistent security and precise representation of enterprise strategical and legal environment are needed, and have to be represented in a common *global security* scheme.

The goal of this paper is to show how Digital Rights and Policy Management (DRM/DPM) driven security can be used in order to offer a global solution tackling these issues.

This paper is structured as follows. Section 2 details how DRM/DPM can provide interoperable security. Some other existing approaches ar discussed in section 3. We present ongoing work in section 4 and we finaly conclude in section 5.

2 Security interoperability through DRM/DPM

This section presents how Digital Rights and Policy Management can enable interoperable security and how their use covers security needs from both strategic and operational point of view.

2.1 Enterprise Digital Rights and Policy Management

With modern Enterprise Information Systems, enterprise processes have to span accross multiple corporate structures. In this context the issue of persistent content protection, rule based access to content and usage metering appear as key requirements. In order to protect themselves, enterprises have to deal with the recurring problem of managing, safeguarding and controlling usage of information assets wherever it resides and especially outside the corporate firewall. The technology addressing this security issue is called Digital Rights Management (DRM) and its strategic managerial dimension is called Digital Policy Management (DPM). As a result, any attempt to access a content protected by DRM requires interpreting associated rules prior to granting or denying the right to do so [2]. Enterprise DRM and Enterprise DPM, are the application of DRM/DPM to the corporate environment in order to protect any kind of corporate informational resource.

One of the major problems that hampered broader and faster adoption of DRM was the lack of standards and the totally incompatible proprietary solutions that were available (e.g., Microsoft, InterTrust, ContentGuard, IBM, etc.). Nevertheless, associations like OMA (Open Mobile Alliance) [3] or MPEG-21 [4] are actively working on DRM standardization and the situation is progressively changing. Recently ISO ratified two sandards developed within MPEG-21 : MPEG-REL (ISO/IEC 21000-5:2004), a Rights Expression Language based on XrML [5], and MPEG-RDD (ISO/IEC 21000-5:2004) which addresses the issue of rights interoperability and semantics through RDD (Rights Data Dictionary). Reference implementations for both MPEG-REL [6] and MPEG-RDD [7] are currently available. Such initiatives are instrumental in this field and represent a prerequisite for broader adoption and interoperability.

2.2 Security meeting strategical and operational needs

As specified earlier, policy management is a key strategic issue for the Enterprise. Information is a corporate asset and is therefore bound to corporate policies. Companies need to be able to define and control who, how, when, what, in what context, and under which condition enterprise resources can be accessed and used at all time. These resources can be of any type like financial statements and reports, design documents, technical specifications, proposals, contracts, legal documents, emails, etc. Moreover, not only *documents* need to be protected. Processes, executable code, raw data, and value added information provided by databases and application servers, dynamically generated and which do not exist statically but are the result of specific queries are also bound to usage rights and policies.

Basic security notions providing confidentiality, authentication, integrity and non repudiation are still needed but not sufficient anymore. Digital Policy Management is of strategic nature and as such must be initiated and driven by corporate managers and not IT / IS people. It has to take into consideration three different but coexisting views : the legal environment, specific regulatory frameworks often sector bound and internal corporate policies.

Enterprise DRM meets Enterprise DPM at this point. It requires to address the issue in an interdisciplinary space between technology and management science, having to apply at the operational level decision taken at the strategical level.

2.3 DRM/DPM driven interoperable and dynamic security

Regards to information systems security, following observations can be made. First, security policies defined at a strategical level have a global validity. Regardless of the way information systems are implemented these policies have to be respected at operational level. Second, as described before, monitoring and tracability ability are an important part of modern information systems security needs. Further, rules defining if a resource (any kind of data or any system functionality) should be accessible or not depend on the context in which the ressource is accessed. As information systems are heterogenous, security interoperability is needed to provide a global view of the security context.

In order to provide more flexible and adaptable security as well as interoperability, traditional access control lists (ACL) could be replaced by DRM/DPM. Unlike ACLs wich are a static definition of access rights, DRM provide dynamic interpretation of rules, granting or denying access to resources depending on the result of a verification process. These rules can define who, how, when, for how long, can access what resource in what context.

Credentials are emited for well defined users or processes by a certified entity, granting access to functionalities and resources of the system. When a ressource is accessed, the process accessing the resource has to provide a set of credentials. The system checks if these credentials match the rules associated to the protected resource and decides to grant access to the resource or not to do it. Credentials can also be dynamically revoked if needed, for instance in case of abusive behavior, or refined to grant access to more resources.

Global security interoperability can be provided using sets of credentials. These can be used to access any resource anywhere on the information system. There is no need for the different systems to understand every credential a process provides. Once the rules mediating access to a resource are set, only relevant credentials have to be recognized. If a credential is not known, the system has to be able to check if this credential is equivalent to a known type of credentials. Such an interoperability of credentials can be achieved using standards like MPEG-RDD.

As stated before, the rules are defined at the strategical level and have to reflect strategical environement evolution. Using DRM/DPM, security can be dynamicaly refined with no need to modify already emited credentials but only by modifying the set of rules affected by the environmental change. Changing the rules has a global impact, avoiding the risk to forget to modify the rights of one given human user or process.

As each credential is affected to a particular process or user, and as the whole security scheme is based on a dynamic verification of credentials towards rules, tracability or even real-time monitoring tools can be easily implemented. Such tracability can be used to capture the context and information flow that has led to a particular access to a resource. The verification engine could then use such information to check higher level rules covering multiple aspects of the system.

3 Other approaches

Two main approaches are in use in order to tackle issues related to the authentication on multiple interconnected enterprise systems : *password synchronization* and *single sign-on* (SSO). None of these approaches tackle security issues in their globality, from the strategic, operational and interoperability point of view.

Password synchronization's goal is to ensure that when a user changes his password on one system, this change will be propagated to all other accounts automatically. The user still has to log into each system independently, but he has only one password to remember. This is a simple way to have enforce users to have unique passwords for all parts of an information system. Such an approach is inherently insecure and definitively not designed for providing global security.

Single sign-on is a mechanism whereby a single action of user authentication and authorization can permit a user to access multiple applications within the network where the user has access permission, without the need to enter multiple passwords. Single sign-on goal is to reduce human error, a major component of systems failure [8]. Existing SSO solutions propose features like the use of strong authentication, such as biometrics, enforcement of password policies, definition of enduser credentials or audit of logins. These solutions mainly focus on securing access to ressources and monitoring actions. As SSO solutions are often used in enterprise application integration solutions (EAI) (like Microsoft BizTalk Server) they do not address the issue of interoperability of different SSO solutions, nor the interoperability with systems not designed for a particular SSO solution.

Some DRM solutions make use of SSO to provide user autentication for DRM (like Fasoo [9]), but they only use DRM to ensure the security of enterprise documents, and not in order to globaly manage enterprise information systems security.

4 Ongoing Work

We are currently working on the modelisation and development of a prototype where a mobile agent platform can interact with a database using DRM (MPEG-REL, MPEG-RDD) to ensure global security. We chosed databases and Mobile Agent Systems [10] domains in order to represent an example of the heterogeneity of modern information systems.

Agents possess a set of credentials that have been emited and signed at agent creation time by a trusted entity. These credentials define the rights an agent possesses as well as in which context these rights apply. Rights can evolve in time and be refined. The ability to access to some resources or behaviours can be dynamicaly restricted. Similarliy, new rights can be obtained and extended, offering the ability to access previously restricted resources.

In parallel, rules are associated to resources provided by databases or by agents. Each access to any type of resource (database entries or agent method / service) triggers dynamic credentials verification at runtime towards existing rules. After verification, access is granted or denied. The scope of a rule is not necessarily a paricular resource. A rule can involve multiple resources and even other rules in orther to represent in which context a process tries to access a resource. Thus, rules can be directly used to specify information flows. Moreover, rules can be dynamically modified to continuously reflect policies defined at the strategical level and thus restrict or widen access to resources.

The overall goal of this prototype is to provide fine grained control on the way resources can be accessed and show how enterprise processes can be controled using DRM/DPM. Further using DRM in a global way induces security interoperability. Indeed, in this context, from the agent perspective there is no difference between the security scheme an agent has to use to access a resource located on a database or access a resource, method, provided by another agent or any other sytem.

When accessing a resource an agent provides matching credentials. The system checks if resources can be provided depending on dynamic interpretation of existing rules. An interesting feature provided by DRM is that there is no need for global identification and management of passwords as such information can be directly encapsulated into provided credentials. Knowing who emited the credentials for whom allows to trace and monitor agent behaviour.

As every possible situation may not be anticipated by legitimate content "right holders", formalizing exaustive rules sets appears to be hardly possible. An interesting approach to explore to tackle unexpected situations is DRM Exceptions Management [11]. DRM Exceptions provide the ability to entities asking for resources to ask for a short lived Exception Licence (as a X.509 Attribute certificate) authorizing them to access needed resources. The licence being emited if and only if the exception is recognized as valid.

Other interesting aspects that have to be considered include, for instance, the ability to define and manage what processes are authorized to do with the data they use, what are the rules associated to generated information, and how and under which conditions, local (resource related) security can be delegated to the processes using them.

Nevertheless, DPM field is currently cruelly lacking models to capture, specify, express, represent and manage policies prior to any technical DRM project and deployment in a corporate environment. Thus, future research effort will also consist in finding ways to fill this gap.

5 Conclusion

Enterprise Information Systems are living heterogeneous entities. They must be able to reflect enterprise's environement and evolve with it. Enterprise information assets are of multiple types. Any kind of document, but also raw data, processes and value added data, taken and processed from existing databases and any combination of existing systems have to be protected.

As such, when designing future information systems, multiple aspects and issues have to be considered . This includes low level security, interoperability, evolutivity, tracability and the ability of enterprise systems to answer strategic questions as well as their ability to be aligned on enterprise's strategic decisions. Integrating all these views in a common viewpoint can be called *global security*.

This paper has presented how Digital Rights and Policy Management can provide *global interoperable security* for enterprise information systems, able to meet strategical security needs and able to implement these needs at the operational level.

References

1. Sarbanes-Oxley: Sarbanes-oxley act. http://www.sarbanes-oxley.com/ (2005)

2. Jean-Henry Morin, Michel Pawlak, M.O.: Enabling technologies for the interoperable enterprise. In: Workshop on Interoperability of Enterprse Systems (INTEREST), ECOOP 2004 - 18th European Conference on Object-Oriented Programming (2004)
3. OMA: "open mobile alliance". http://www.openmobilealliance.org (2005)
4. Moving Pictures Experts Group: "mpeg-21 overview v.5". http://www.chiariglione.org/mpeg/standards/mpeg-21/mpeg-21.htm (2002)
5. "ContentGuard": "extensible rigths markup language (xrml), version 2.0". http://www.xrml.org (2005)
6. ContentGuard: "mpeg-rel sdk". http://www.contentguard.com (2005)
7. RightsCom: "rights data dictionary". http://www.rightscom.com (2005)
8. Opengroup: Single sign-on. http://www.opengroup.org/ (2005)
9. Fasoo.com: "fasoo enterprise drm". http://www.fasoo.com (2005)
10. Vitek, J., Tschudin, C., eds.: Mobile Object Systems: Towards The programmable Internet. Springer Verlag, LNCS 1222 (1997)
11. Morin, J.H.: A credential based approach to managing exceptions in digital rights management systems, CCNC/CES 2005 Workshop on Digital Rights Management Impact on Consumer Communication (2005)

Enabling inter-enterprise collaboration in open distributed systems

Toni Ruokolainen

Dept. of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FI-00014 UNIVERSITY OF HELSINKI
FINLAND
toni.ruokolainen@cs.helsinki.fi

Abstract. One of the most important infrastructure services of tomorrow's enterprise computing platforms are the services needed for interoperability analysis and management of typing disciplines. In my research I will study the characteristics of business collaboration networks which use business services and their compositions to reach its business goal. This research will begin with an analysis resulting in thorough understanding of the nature of interoperability conflicts inherent in business networks. From these results a typing disciple for business services and networks based on a formal theoretical framework shall be formulated. Resulting type system is applied in practise as type repository functionality for web-Pilarcos platform.

Globalisation of markets, considerably stimulated by the development of ICT technology, introduces new kinds of demands for enterprises willing to stay competitive. Viability of an enterprise is not determined by its core resources or competencies but also by its ability to engage business with various partners, possibly in quite different business domains. An increasing amount enterprises' income is made inside collaboration networks and as a part of inter-enterprise value and supply chains.

Competition has shifted from rivalry between distinct enterprises to competition between business networks. Business network is a collaboration between enterprises with a certain business goal. In free markets business network which provides more valuable services or reaches its goals with lower costs will outlast those with lesser value or higher costs. To maintain its competence, a business network must be allowed subtle changes or evolution to reflect changing business needs and environments. Evolution of business network might be as simple as changing previous partner to a more suitable one or even changing business goals during a collaboration.

Flexible and dynamic collaborations shall be established even for short-term usage. This should be possible without considerable development or maintenance costs. Tomorrow's enterprise information systems and especially the supporting infrastructure must provide means for establishment and management of collaboration networks in an environment with autonomic and heterogeneous agents.

Enterprises must have a possibility to maintain and adapt their own services as needed. An enterprise wants to exploit technological developments and refined business processes to retain its competitive edge. To support this autonomy we cannot presuppose that distinct services will be implemented using same technological platform or that they would always behave the same. Business services offered by enterprises have independent development cycles and lifetimes. Enterprises are also autonomic administration domains with their internal policies. In this kind of operational environment static verification of interoperability is insufficient.

Collaboration model we have taken in web-Pilarcos project is quite similar to those in virtual organisation and workflow systems, such as WISE or CrossFlow [1, 4]: contracts stating the responsibilities and behavioural properties of participants are used as the basis for business network operation. We follow an unified model where each participant in a collaboration may be using different execution platforms and technologies as long as they comply to the restrictions and behavioural patterns of the collaboration contract.

Business network descriptions are used to define the characteristics of an inter-enterprise community, such as topology of the community, processes used between participants and community specific functional and non-functional constraints and properties. Business processes are not used for execution of enterprise services as in traditional workflow systems, but as behavioural descriptions needed for static verification of service interoperability and dynamic monitoring of conformance between the behavioural contract and actual service behaviours.

Interoperability of business services is an issue that must be supported during design, implementation and operation of business collaboration networks and business services. Static verification of interoperability should be enforced by the mechanisms and disciplines used to develop, publish and manage business services and networks. When a new service is published to a public service trading system its behavioural properties and especially its conformance to the claimed service type must be verified. Type repositories are used to enforce these verification obligations and correct use of typing disciplines. During publication of new business collaboration models behavioural properties and logical integrity of these descriptions must be verified.

Process algebras, especially different forms of typed π-calculus [6, 5] can be used as theoretical framework for development of business service typing discipline and interoperability verification framework. To achieve a typing system where a service type can be used in different business networks, interface descriptions should be equipped with process descriptions that define bilateral behavioural patterns. One of the most greatest challenges will be the development of a typing discipline which provides "sufficiently complete" proof of service interoperability, yet supporting independent service and business network evolution.

Composition of service types into roles, the functional units of business service networks, must be supported by the type system in a such way that verification of composite behaviour remains a relatively light procedure. A type effect system mechanism similar to correspondence assertions introduced in [2] could be used to separate the aspects of communication and coordination in business services and their compositions. Another interesting possibility could be use of some other logic, e.g. tile logic [3], to express and separate coordination and communication needs of service compositions.

Static verification done during development and publication phase of business services is not sufficient in our context with autonomic administration domains. For example internal policies not visible outside an enterprise may evolve during operation of a community, causing changes to functional or non-functional properties of its services. For this reason runtime monitoring and verification must be executed during operation of business networks.

My research will concentrate on research of interoperability support and management in collaboration communities involving autonomic business services. First task of my research will be a categorisation on the nature of interoperability conflicts: what are their causes and how could they be prevented or observed before they happen. Results of this analysis will be used in formulation of a typing discipline that shall be used for description of business services and networks.

Typing discipline must allow evolution and reuse of business services in a flexible manner. For this purpose I think it is necessary to separate aspects of communication and coordination such that these both aspects can be analysed separately and their properties and structures changed as independently as possible. Communication and coordination should be treated as orthogonal properties, as much as possible, the first one described in the interfaces of business services and the second one described mainly in business networks as "glue" composing distinct service types into business roles.

The theoretical framework must however support propagation of behavioural effects from the coordination dimension to the communication dimension and vice versa such that correct operation of role compositions can be verified. Theories and mechanisms of typed π-calculus and e.g. type and effect systems and model checking shall be studied for this purpose. Session types [5] are now used for the description of bilateral service protocols, but its applicability as part of service typing discipline is yet to be researched. The theoretical framework and the resulting typing discipline shall be tested in practise during my research: a type repository functionality is used as part of web-Pilarcos platform to ensure service interoperability.

References

1. G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, and N. Weiler. Wise: Business to business e-commerce. In *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, page 132. IEEE Computer Society, 1999.
2. E. Bonelli, A. Compagnoni, and E. Gunter. Correspondence assertions for process synchronization in concurrent communications. In *FOCLASA 2003, the Foundations of Coordination Languages and Software Architectures*, volume 97 of *Electronic Notes in Theoretical Computer Science*, pages 175–195. Elsevier, July 2004.

3. R. Bruni and U. Montanari. Dynamic connectors for concurrency. *Theor. Comput. Sci.*, 281(1-2):131–176, 2002.
4. Y. Hoffner, H. Ludwig, P. Grefen, and K. Aberer. Crossflow: integrating workflow management and electronic commerce. *SIGecom Exch.*, 2(1):1–10, 2001.
5. K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In *Proceedings of the 7th European Symposium on Programming*, pages 122–138. Springer-Verlag, 1998.
6. N. Kobayashi. *Formal Methods at the Crossroads: From Panacea to Foundational Support*, volume 2757 of *Lecture Notes in Computer Science*, chapter Type Systems for Concurrent Programs. Springer-Verlag, 2003.

Matchmaking in Heterogeneous Environments

Dong Cheng

LORIA-University Henri Poincaré Nancy 1
BP 239, 54506 Vandoeuvre Lès Nancy CEDEX(F)
Email: cheng@loria.fr

Abstract. In more and more service domains, composite services will be necessary, as application ranging from search engines (such as Google, Yahoo, etc.), to more general applications, like cooperative and distributed applications or e-business and e-commerce applications. In a lot of situations, the usable information is recorded by different persons, at different times, in different languages, on different fields. The composite service programme must have the capability to process the heterogeneous information. In this paper we focus on the problem of *mismatch* in knowledge representation languages. Specifically, we have implemented a matchmaking service in a homogeneous knowledge representation environment (using *Description Logics*). Based on this experience we show an approach for measuring *mismatch* in an heterogeneous knowledge representation environment.

Keywords: *heterogeneous system*, *mismatching*, *knowledge representation*

1 Introduction

The propagation of the network has led to increasing need for interoperability between heterogeneous systems and services. These latest developments in data network domain are new challenges for interoperability technology, as attested by the tremendous work about Semantic Web [1]. However, in some situations the user who poses a query expects a unified answer from a system that is more than a "yes/no" answer (i.e. either the system finds out answers to a given query or fails). The unified answer will be a composite answer by several heterogeneous systems. In this condition, we must match the information that was created by different programmers, at different times, in different languages, and all these things are described in different *knowledge representation technologies*. Examples of applications of our work may be searching for Web services that offer given functions, searching for a component in the context of component-based design and component-based programming, searching for a business partner with a given expertise, looking for an employee whose records and expertise satisfy a given position profile...

In this paper, we highlight a *mismatching* approach in heterogenous environment where the systems record the information in different knowledge representation technologies. A significant originality of our approach resides in the type of answers we aim at providing. Indeed, when there is no unique entity to satisfy the search criteria, the systems attempt to determine a composite answer that satisfy the criteria.

The presentation is structured as follows. In section 2, we expose some motivations and possible application domains. In section 3 we review the federated mediator-based architecture, and some theory models, and concepts of composite answers to this architecture. Section 4 briefly introduces some knowledge representation technology. And we put conclusions and remarks are in section 5.

2 MOTIVATION

Schema matching is a classic problem in many database and data integration applications. Several approaches that were used in many application domains, such as schema integration, data warehouse, e-commerce and semantic query processing, have been proposed in the last decade [2]. In particular, discussions are generally limited in combination of multiple matchmaker on schema and instance level, element and structure level, and language and constraint based. Among these the *mismatch* is a common issue of interoperability in composite applications, such as mismatches in database schemas or semantic query processing [3].

In the semantic query processing that we propose, the whole query, Q is described by a concept (or class) that is written in a formulable knowledge representation language $\mathcal{L}_{Query}$. Q is sent to the matchmaker asking for individuals fitted with a given set of capabilities. The matchmaker returns an answer $\mathcal{Q}_{Answer}$ which is also expressed in a formulable knowledge representation language, denoted as $\mathcal{L}_{Answer}$. Let $\mathcal{Q}_{Answer} \doteq \mathcal{Q}_{Satisfaction}$ denote the fact that a query Q is fully satisfied, and let $\mathcal{Q}_{Answer} \doteq \{\mathcal{Q}_{Satisfaction}, \mathcal{Q}_{Complement}\}$ denote the fact that a query is partly satisfied. $\mathcal{Q}_{Complement}$ is a formularized description of mismatch that describes "what is missing" to the individuals which are described in $\mathcal{Q}_{Satisfaction}$, to satisfy Q, that means to determine what part of the query is not satisfied by the found individuals. This part, together with the original query, are transmitted to another information service by matchmaker.

Let us now elaborate more on the conceptual framework using, as an example, an application on *Web Services* (see Figure 1). A query Q, which is expressed as a sentence of a query language $\mathcal{L}_{Query}$, is sent to a matchmaker. The matchmaker may find n offerings which are expressed as sentences of an answering language $\mathcal{L}_{Answer}$. So we get a set of answers $\{\mathcal{Q}_{Answer\ 1}, \ \ldots, \mathcal{Q}_{Answer\ n}\}$, all written as
$\{\{\mathcal{Q}_{Satisfaction\ 1}, \mathcal{Q}_{Complement\ 1}\}, \ \ldots, \{\mathcal{Q}_{Satisfaction\ n}, \mathcal{Q}_{Complement\ n}\}\}$. In this work we focus on interoperability in heterogeneous environments. In such environments, a query Q and an answer $\mathcal{Q}_{Answer}$ may be represented in different languages. Two non exclusive differences may be considered. The first one may be lexical difference; Q and $\mathcal{Q}_{Answer}$ do not use the same vocabulary. The second difference may concern the languages of Q and $\mathcal{Q}_{Answer}$ (for example: Q is expressed in Description Logic and $\mathcal{Q}_{Answer}$ in F-Logic, Q is expressed in

English and Q_{Answer} in Italian, etc.). Concerning the lexical level, numerous lexical translation approaches were introduced [5] [6] [7]. These studies on the lexical level will contribute to the determination of an answer composed from "fragments" expressed in different knowledge representation languages (as *Description Logics, Conceptual Graph*, etc.).

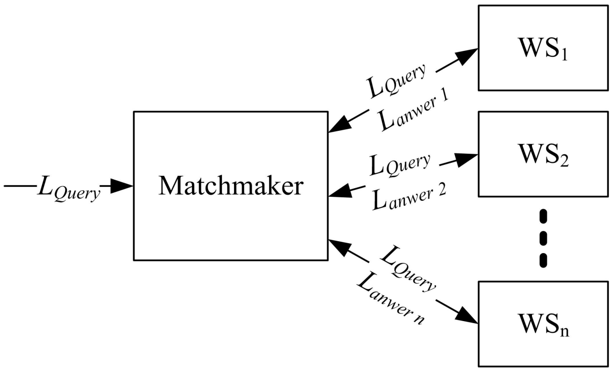


Fig. 1. The Model of Matchmaker

3 A Mediator-based architecture for answer composition

In our work a mediator-based architecture has been adopted and is described in [4]. It is a dynamic discovery of services or capabilities an "entity" offers. It is very similar to the notion of *discovery agency* in the Web services architecture. In this architecture, an "entity", called *exporter*, publishes (*tells*) its capabilities at one or more mediators sites (see figure 2). Entities, called *importers*, send requests (*queries*) to the mediator asking for exporters fitted with a given set of capabilities.

Several approaches, which are mediator-based, i.e. they are distributed and intelligent production systems, have been proposed over the last decade. A single Mediator is designed to offer an adequate level of decision-making integration taking integration, it takes into account the effort needed for the integration of heterogeneous computer systems [8]. The Conflict Resolution Environment for Autonomous Mediation (CREAM) system has been implemented, it provides various user groups with an integrated and collaborative facility to achieve semantic interoperability among participating heterogeneous information sources [9]. The KRAFT (Knowledge Reuse And Fusion/Transformation) architecture provides a generic infrastructure for knowledge management applications. It supports virtual organization using mediator agents [10]. When it was applied to business-to-business electronic commerce, the mediators allows partners to exchange rich business information.

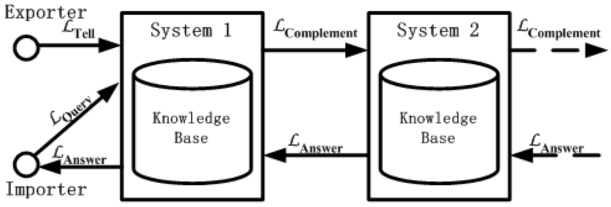


Fig. 2. The Mediator-based Architecture

In federated mediators, a cooperation environment, we can see the query as being addressed to "the union" of the federated mediators' knowledge bases. Concretely, this union is explored from "near to near" within the federation, that means from a mediator to an other. Satisfying the query falls into different cases [4]:

- Case1: There exist exporters that exactly satisfy the query;
- Case2: There exist exporters that fully satisfy the query, but their capabilities are wider than those requested;
- Case3: No single exporter fully satisfies the query, but when "combining" or composing capabilities from different exporters, one can fully satisfy the query;
- Case4: Neither a single exporter nor multiple exporters satisfy the query, but there exist some exporters that partly satisfy the query;
- Case5: No single exporter nor several exporters fully or partly satisfy the query.

One notices, that cases 3 and 4 are cases where a composite answer is returned. In the case 4, we have to determine "what is missing?", *Complement Concept*, to the individuals to satisfy Q, which means to determine what part of the query is not satisfied by the found individuals. Furthermore, in these cases 2, 3 and 4, we need to notice "what is superfluous?".

An approach to find out "the best" answer was proposed using Description Logics [11]. The determination of the complement is based on the subsumption relationship using a *Normalize-Compare* process. The aim of the normalization is to put defined concepts to be compared, let say A and B, under a conjunctive form: $A = (\text{and } A_1, A_2, A_3, \ldots, A_n)$ and $B = (\text{and } B_1, B_2, B_3, \ldots, B_m)$. Once normalized, two concepts can be easily compared to check wether the subsumption relationship holds between pairs of them or not: giving $A = (\text{ and } A_1, A_2, A_3, \ldots, A_n)$ and $B = (\text{ and } B_1, B_2, B_3, \ldots, B_m)$, the test "does the concept A subsume the concept B?" returns "true", if and only if $\forall A_i$ (i $\in$ 1,...,n) $\exists$ B_j (j $\in$ 1,...,m) such that $B_j \sqsubseteq A_i$. The implementation of this process uses an array of Boolean (called "Table_Of_Test" further) to record the results of the subsumption relationship evaluation. In the figure 3, $C_1, C_2, C_3, \cdots, C_n$ denote the query concept under its normal form and $D_1, D_2, D_3, \ldots, D_m$ denotes the concepts "known from" the mediators, i.e. every D_j has to be viewed under its normal form $D_j^1, D_j^2, \cdots, D_j^{nj}$. Then $Table\_Of\_Test[D_j, C_i] = true$ means that $D_j^i \sqsubseteq C_i$. When the value returned by the function $Subsumes(C, D_j)$ is "false" (i.e. the concept D_j does not fully satisfy the concept C.), therefore we need to determine a possible complement of D_j relatively to C. Using the Table_Of_Test it is easy to get the complement of the concept D_j relatively to the concept C: $Comp(C, D_j) = and_{k=1}^{r} C_k | TableOfTest[D_j, C_k] = false$[1]. That means that the complement is given by the conjunction of all the atomic concepts for which the corresponding values in the "Table Of Test" are "false".

	C_1	C_2	...	C_n
D_1	False	False	...	True
D_2	False	True	...	True
...	...	...	...	...
D_m	False	False	...	False
ORoD	False	True	...	True

ORoS	ANDoS
True	False

Fig. 3. The "Table Of Test"

The composition of the truth values determines the cases of satisfaction. Consider a table $ORoD[1..n]$ as $ORoD[i] = \bigvee_{j=1}^{m} T[D_j, C_i]$. $ORoD[i] = true$ means that the concept C_i is satisfied by at least a D_k. If the conjunction of the values of $ORoD$, noted $ANDoS$, is true (i.e. $\bigwedge_{i=1}^{n} ORoD[i] = True$), it means that all the C_is are satisfied and therefore the query. When $ANDoS$ is false, the logical disjunction of the values of $ORoD$, noted $ORoS$, enables to determine a possible partial satisfaction: if $ORoS = \bigvee_{i=1}^{n} ORoD[i] = True$, it means that there exist some C_k that are satisfied. If both $ORoS$ and $ANDoS$ are false then no atomic concept D_j^k $(j \in 1..m)$ satisfies a C_i. The figure 4 summarizes this discussion (in this figure, X, $\top$, and $\perp$ respectively denote a concept, the TOP concept and the BOTTOM concept; *MSSC* and *MGSC* respectively denote the Most Specific Subsuming Concept and the Most General Subsumed Concept; finally the numbers in the CASE column refers to the satisfaction cases listed).

An experimental platform has been developed in *Java* where services, like testing the subsomption relationship, determining the complement of a concept and computing a composite answer, have been implemented. These services are accessed through a *Web Server* in a "classical" Web client/server architecture. The services accept DL concepts written in DAML+OIL [12], an ontology language in *XML*. The DL concepts are encoded in *XML* and transmitted to the *Web Server* who in turn transmits them to the appropriate mediator service. Then a normalization class transforms them into *Java* objects and an experimental knowledge base, described in XML, is loaded and normalized into a

[1] In the following, we will use T[i, j] instead of TableOfTest[i, j]

ORoS	ANDoS	MSSC	MGSC	CASE
True	True	X	X	1
True	True	X	$\perp$	2
True	True	$\top$	$\perp$	3
True	False	$\top$	$\perp$	4
False	False	$\top$	$\perp$	5

Fig. 4. The analysys of the satisfaction case

TBox object when the service is started. All the algorithms of the mediator's services are implemented in the *TBox* class. The services' outputs are also encoded in *XML*. XML is also used to exchange information and concepts between the mediators when mediators' cooperation is needed. In the current status of the implementation, a mediator "discovers" an other mediator using a static mediator address list. More complex and dynamic discovery techniques will be supported in the coming versions. Moreover, we deliberately ignored the search of the actual *individuals* (*ABox*) that satisfy a query, i.e. in the current implementation, only *TBoxes* are considered.

In this service, the *Complement Concept* is under a conjunctive form: $C = (\ and\ C_1, C_2, \ldots, C_n)$. Each C_i is an *atomic concept* where each *atomic concept* is identified by a *term*. The *term* is supposed to exist in a common lexical ontological dictionary (such as WordNet [13], EuroWordNet [5]). Moreover, we suppose facilities for mapping between terms. (Some term mapping approaches in lexical level are described [7]). In our work, heterogeneous systems cooperation thanks to the *complement* concept which is under the simplest form, *conjunction*. The conjunctive form is adopted in most knowledge representation languages, so the *complement concept* may be understood by the other partner systems. These systems may be in different *knowledge representation* languages.

4 Knowledge Representation

Knowledge Representation is a substitute for the thing itself, it must be a set of ontological commitments and a theory of intelligent reasoning [14]. In our work, we adopted *Description Logics* (abbr., DLs) [15] that is a family of knowledge representation languages where a description of a world is built using *concepts*, *roles* and *individuals*. A concept is specified thanks to a structured description that is built giving *constructors* that introduce the roles associated with the concept and possible restrictions associated with some roles. Otherwise, *Frame Logic* (abbr., F-logic) is a formalism that is capable of representing virtually all aspects of what is known as the object-oriented paradigm [16]. *Conceptual Graphs* (abbr., CGs) are a system of logic based on the existential graphs that serve as an intermediate language for translating computer-oriented formalisms to and from natural languages [17]. *Conceptual Graph Interchange Form* (abbr., CGIF) was proposed in CG Standard by John F.Sowa, that represents CGs' syntax and semantics in machine-readable character strings. Of course, natural languages are the most important knowledge representation languages. But it usually is not well-formal, we usually use some *controlled natural languages* to knowledge representation, such as *Common Logic Controlled English* (abbr., CLCE)(by John F. Sowa), that is a formal language with an English-like syntax. Anyone who can read ordinary English can read sentences in CLCE, the sentence that supports full first-order logic is written within its syntactic and semantic limitations. In this work, we draw a lattice of knowledge representation languages (see Figure 4).

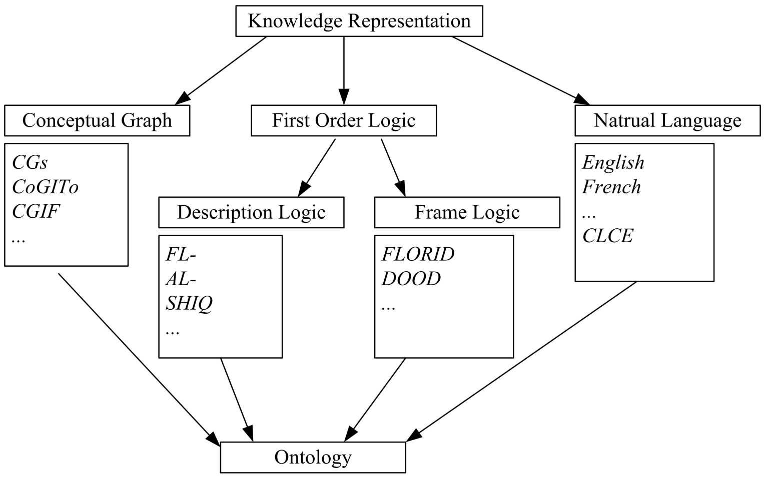


Fig. 5. A lattice of knowledge representation languages

We write a sentence, “The cat is on a mat.”, in five kinds of knowledge representation languages (see figure 6). We may find the three terms (CAT, MAT and ON) in each sentences . From a mathematical view, CAT and MAT are two sets of element, ON is a relation between the two sets. In addition, *universal quantifier*(every, all or ∀) and *existential quantifier* (exists or ∃) may be found in some sentences. The mathematical background of knowledge representation was particularized in [18]. Several approaches that translate the formularized sentence between first order logic with other representation languages were introduced in [19].

In [19] the “term” is the most important component in most knowledge representation languages, which identify some sets of element and relationship between sets. Elementary set theory is used to define basic mathematical structures, which represent collections of abstract objects. In mathematics, sets are usually composed of numbers or points, but we more often talk about sets of persons, animals and all in knowledge representation domain. the *relation* is another important concept in knowledge representation, a *relation* is a function

CLCE: Every cat is on a mat.
Predicate Logic: $(\forall x : Cat)\ (\exists y : Mat)\ On(x, y)$
CGIF: (On [Cat:@every] [Mat])
DLs: (all (some on MAT) CAT)
F-Logic: CAT[on $\rightarrow$ MAT]

Fig. 6. Five types of Knowledge Representation

of one or more arguments whose range is the set of *truth values* {true, false}. Profit from the same mathematical background, we may define the *Complement* concept in different knowledge representation languages.

5 The Way Forward?

We believe that heterogeneous systems may interoperate by virtue of match-making approach. We believe that the hard problems do not go away even if we solve low-level issues such as natural language analysis, complex term mapping, and ontologies integration. Further, we suggest that the root causes can be understood better in terms of *mismatch*.

What can be done about this? Based on our initial experience on homogeneous knowledge representation technology and common mathematical background of heterogeneous knowledge representation technology, two interrelated approaches paid attentions to our work. The first is to find some translation approaches between any two knowledge representation languages. The second is to measure the mismatch, then describe it in a formularized language, and find the algorithm to get a "best" composite answer.

In the future, we may consider to design and implement a platform, where heterogeneous systems will cooperate in more complex logical constructions in different knowledge representation technologies.

References

1. W3C: Semantic Web (2003) http://www.w3.org/2001/sw.
2. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases **10** (2001) 334–350
3. Garlan, D., Allen, R., Ockerbloom, J.: Architectural mismatch, or, why it's hard to build systems out of existing parts. In: Proceedings of the 17th International Conference on Software Engineering, Seattle, Washington (1995) 179–185
4. Boudjlida, N.: A Mediator-Based Architecture for Capability Management. In Hamza, M., ed.: Proceedings of the 6th International Conference on Software Engineering and Applications, SEA 2002, MIT, Cambridge, MA (2002) 45–50
5. Piek, V.: Eurowordnet: A multilingual database with lexical semantic networks. EuroWordNet General Document **3** (1999)

6. Ahrens, K., Chung, S.F., Huang, C.R.: Conceptual metaphors: Ontology-based representation and corpora driven mapping principles. In John Barnden, Sheila Glasbey, M.L., Wallington, A., eds.: Proceedings of the ACL 2003 Workshop on the Lexicon and Figurative Language. (2003) 35–41
7. Niles, I., Pease, A.: Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In: Proceedings of the IEEE International Conference on Information and Knowledge Engineering. (2003) 412–416
8. H.K. Tönshoff H.K., I. Seilonen, G.T., P.Leitado: A mediator-based approach for decentralised production planning, scheduling and monitoring. Proceedings of CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (2000) 113–118
9. Sudha Ram, J.P., Hwang, Y.: Cream: A mediator based environment for modelling and accessing distributed information on the web. In: Nineteenth British Nantional Conference on Database(BNCOD2002). (2002) 58–61
10. Preece, A.: A mediator-based infrastructure for virtual organisations. In Agents-2001 Workshop on Intelligent Agents in B2B E-Commerce (2001)
11. Cheng, D., Nacer, B.: Federated mediators for query composite answers. In: 6th International Conference on Enterprise Information Systems - ICEIS'2004, Porto, Portugal. Volume 4. (2004) 170–175
12. Horrocks, I.: DAML+OIL: a description logic for the semantic web. IEEE Data Engineering Bulletin **25** (2002) 4–9
13. Miller, G.A., al.: Wordnet 2.0 (2003) http://www.cogsci.princeton.edu/ wn/.
14. Randall Davis, H.S., Szolovits, P.: What is a knowledge representation? AI Magazine **14** (1993) 1993
15. DL-org: Description logics (2003) http//dl.kr.org/.
16. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. Journal of the Association for Computing Machinery **42** (1995) 741–843
17. Chein, M., Genest, D.: Cgs applications: Where are we 7 years after the first iccs? In Ganter, B., W.Mineau, G., eds.: Conceptual Structures: Logical, Linguistic, and Computational Issues. Volume 1867., Darmstadt, Germany, ICCS 2000, Springer (2000) 127–139
18. Sowa, J.F.: Mathematical background of knowledge representation (2003) http://www.jfsowa.com/logic/math.htm.
19. Amati, G., Ounis, I.: Conceptual graphs and first order logic. The Computer Journal **43** (2000) 1–12

Information Security in a Heterogeneous Healthcare Domain

Rose-Mharie Åhlfeldt

School of Humanities and Informatics
University of Skövde
P.O. Box 408, S-541 28 Skövde
SWEDEN
rose-mharie.ahlfeldt@his.se

1 Background

Healthcare is an information-intensive activity involving the collection, communication and display of large amounts of information. This information is highly sensitive and most countries have special legislation to prevent its misuse. Hence, it is natural to use the support of computers in order to efficiently improve such an information-intensive organization. The increased use of computers for handling the information also gives access to information held in databases in a way that was previously impossible [21]. Swedish healthcare has gone through an efficiency improvement the last few years but it will also face major challenges and changes in the years to come.

With a new holistic view on the care offer and with a totally unbroken care-chain from the emergency treatment and primary care to rehabilitation and home healthcare and home services of the municipalities, new ways of working and new technology solutions are required. To decrease both time of care and waiting-time, and to increase the quality of the care, the information must be available for the care performers who need it, as quickly as possible, no matter where they are in the care-chain. One condition is that the information is current and correct, which is not always the case when the flood of information in the care stream is without computer support [23].

The main thing is to achieve two important aims concerning information security in healthcare. The first is to reach a high level of patient security, i.e. to give patients opportunities to the best care with right information in right time. The other aim is to reach a high level of patient privacy; i.e. to protect patients from that sensitive information is distributed to unauthorized persons. These aims are hard to reach together. Often this occurs at the expense of either one or the other aim. For that reason a balance between these aims is necessary when the work of information security within healthcare will be discussed [24].

This paper is divided in two parts. The first will give a background to the area of information security in healthcare and problems related to it. The second part will discuss the research problem area and address some research questions.

1.1 The patient record

One of the most central units in care information is the patient record. The traditional paper-based patient record used in a clinical setting generally contains the notes of clinicians and other care providers. Dahlin and Arnesjö [5] declare that the overall purpose of the patient record is to facilitate and support that the patient will be given excellent and secure care. This purpose presumes that record data is reliable and available when needed in the care and understandable for the care performers.

In the healthcare domain, there exist a large number of computerized patient records, so called electronical medical records (EMR). Many of them are single system and do not have the possibility to interact with each other. Just in one hospital there can exist hundreds of different EMR systems. These systems are autonomous and they have different purposes and functions for different kinds of businesses in healthcare. There are problems to exchange information between these systems within the same organization. The problems will not decrease when patient information is needed between different healthcare organizations, as well as primary care to municipalities or municipalities to hospitals or vice versa.

Since the EMR systems include sensitive patient information, the security aspects concerning the managing of these systems are very important. It is not just a piece of paper to take care of. The information is stored digitalized and is included in different systems. From a user's perspective, this can be hard to survey and control. New qualifications in the healthcare domain are required to manage the information security in a sufficient way [24].

1.2 Information Security

It is out of the scope of this paper to give an exact definition of the large number of security terms that exist in the literature. However, since the term information security is used in the paper, it is important do define what the term stands for. Information security is a broad term covering security issues in all kinds of information processing. It includes both technical and administrative security (Figure 1). According to SIS [9] information security is defined as security concerning information assets regarding ability to maintain a desired confidentiality, integrity and availability (also accountability and non repudiation are included). SITHS [17] defines the term information security as the collected effect of measures to minimize the risks addressed for the availability, confidentiality, integrity and accountability of information. In a computerized environment it is easy to focus more on technical measures and functions. It has still been shown that in order to get a sufficient level of information security in the organizations, a structured way of working is required both in planning and implementation of the security work. Furthermore, this is also important for the specific user in its daily work. If not the administrative security is function properly, the technical security will fall short for the security work at whole.

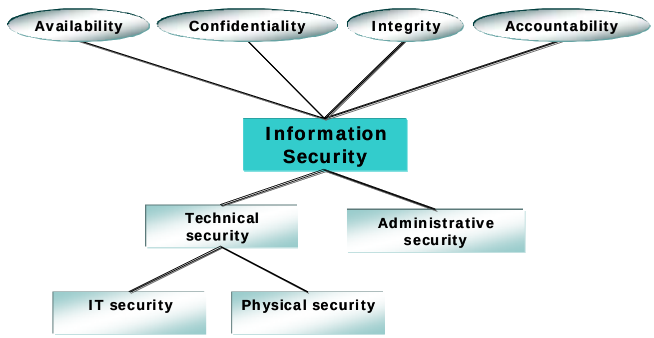


Fig. 1. Information Security Model

Information systems in healthcare are important systems for the society and information security in healthcare is therefore an important element to take into account. Most of the security problems are generic and concerns information systems in general. The Swedish Emergency Management Agency (SEMA) has in their report [16] pointed out some important conclusions concerning all important information systems in the Swedish society.

- Lack of a comprehensive view in information security
- There is a need of a base level of security in important society systems
- Internet should be classified as a critical infrastructure
- There is a need of a political comprehensive view concerning risks, crimes and threats related to information security
- The main threats are unintentional threats
- There is a great need of qualified education

These conclusions are generic for information security in the Swedish society, which imply that the healthcare sector also has to be brought up against with these problems.

Healthcare has a challenging task to deal with an increasing number of citizens, getting both older and sicker [3]. There are reports, showing that if the society is going to succeed with the future's need of care and with limited resources available, there is a need of effectiveness with IT as a support. The healthcare business claims that IT-support is necessary to preserve and improve service, availability and quality in care they are responsible for [3]. Therefore, to keep and improve information security in healthcare is a challenging task. According to SEMA [16] the need of research and studies in this area is obvious.

In the following sections, a number of current problems will be described more in detail. These problems are produced from earlier work and projects along with literature surveys. The describing problems are then going to be the base for the chose of my research problem and the continuing research studies in the area.

1.3 The integration problem

When IT-systems are going to be integrated, and when different healthcare organizations are going to share information with each other, it also makes new demands on information security. Not only the own organization must have a satisfying protection. Instead, the security boundaries are going to expand and the availability to sensitive information is increased. It makes new and bigger demands on security concerning networks, patients' privacy, access control systems etc. Furthermore, the administrative security gets further elements to take into consideration. Co-operated healthcare organizations' working processes and routines must also be considered.

Many healthcare organizations consider that the most important healthcare processes are the patient process [17]. This is the process that from the patient's perspective implies that different healthcare performers must co-operate together with the patient to increase the patient's quality of life. There is a need of a holistic view of information security and cooperative infrastructure for information security but also needs of techniques concerning strong authentication as well as the derived services as authorization, access controls, accountability, confidentiality etc. The SITHS project has created an infrastructure and model to meet all the demands mentioned above. Despite that, the healthcare organizations have not yet adopted this. Instead, they have still a long way to go. One of the reasons for the delay is limited economical resources. The projects also show that there is a need for further studies and research in this area [3].

The demands, mentioned above, are not unique for Swedish healthcare compared with other countries. Even if tradition and legal aspects are different between countries, the main problem is the same. Strong authentications, derived services as authorisation, access controls, accountability, integrity and confidentiality are importunate demands to provide [1], [2], [4,], [26].

When healthcare systems are going to be integrated the interest of a process oriented approach for healthcare information systems is increased [15]. Here, not only the technical and organizational demands on security are increased. Also, the patient privacy is important to keep protected when the patient processes will be extended in distributed systems including sensitive information. Louwerse [12] proclaims that with distributed systems in healthcare we need better awareness, better procedures, better software, and a more centralised approach or systems management.

One way for the healthcare units to be process oriented and integrated with other units' processes and IT systems is to introduce a process manager, which visualises and executes the communication between different IT systems; realised by using graphical and executable process models. The process manager also communicates directly with the healthcare personnel via desktop computers and mobile devices. VITA Nova [14] was a Swedish project which introduced a prototype system for healthcare processes based on a process manager. The result shows e.g. when introducing a process manager in healthcare it requires ways to deal with security issues (security, ethics, and legality). The healthcare units also show large differences in security awareness and IT maturity.

Transferring information between organizations must be done by designing solutions that are satisfactory from the perspectives of the patient and the patient's relatives as well as from a legal perspective. If healthcare in the future will be able to integrate its business processes with those of other care providers, and if transfer of patient information will be satisfactorily performed from a security perspective, routines for the minimum level of security have to be designed, documented and implemented jointly by all involved units [14].

1.4 The lack of security awareness and education

Ever since IT made its entrance to the healthcare sector, the lack of education has been more and more obvious. It's not only lack of education in the security area. Also education concerning teaching new information systems irrespective of their intentions for patient information, administration or network systems has been conspicuous by one's absence. Earlier work has shown that the users only got a few days to learn the system [29]. Depending on the minimum of time scheduled for education of new IT systems, there is neither space to keep up with nor any change to give priority to security education. This lead to users of IT systems in healthcare not having any adequate security education and this also implies that there are lacks of users' security awareness [6].

In the third report the SITHS project declares that the most important element to get the information security to work in an optimal way is education [18]. It is in the education existing threats, risks and possibilities can be pointed out. The SITHS proclaims that this is the way to build security awareness. Measures of security, no matter how technically advanced they are, can never replace the knowledge of the staff and the attitude to the security work. Furthermore, it is enormously important that ethics and moral is kept on a high level. The healthcare process rests on an information flow where the security solutions are needed all the time [18].

It is not only the users and workers in the healthcare sector who need education. Also the healthcare management needs education in information security. Katsikas [11] claim that "the most important pillar upon which any serious effort towards introducing and enforcing security in health information systems is based, is the level of awareness that those responsible for managing the effort have. Thus, the issue of proper awareness, training or education of healthcare management is of paramount importance". Different authors claim that education in information security is an important element, not only for users. Also the healthcare management need education in information security ([17], [18], [11], [29].

The lack of education is often related to the absence of established IT-security strategies and policy issues. Established IT-strategies or information security policies are not so common for the healthcare organisations nowadays [8], [7], [29]. This is an on-going work in the county councils and regions in Sweden. Some have accomplished more than others. Municipalities have the same situation but with bigger variations. There are municipalities with established strategies and policies whereas other municipalities just have started up with the work. According to SWEDAC [22], accredited organizations, for instance laboratories, have been forced to establish these documents which imply that the security work is satisfied and users' education in security issues is an on-going process. Furthermore, this situation has increased the users' security awareness on the whole, not just when they work with computerized systems. This will be in accordance to the result from the VITA Nova project [27]. There is a variation between the different healthcare units depending on how far the organizations have got in their establishing of security policies. The accredited organizations which have demands on themselves to establish security policies states that the employees have a much better security awareness than those organizations which have not come so far yet [14].

Even if strategies and policies are established, it happens that they are inconsistent. This is related to the own organization with different areas and different departments but also when patient information is going to be exchanged over different organisation boundaries. Inconsistent policies and procedures can lead to frustration, confusion and potentially even harm to patients. This is exemplified by differences in organisations' policy towards transmission of patient information by, for instance, facsimile. An organisation attempting to apply more restricted use of data transmissions is faced with complaints from other organisations with more lenient policies whose staff are frustrated that they can not send or receive patient information by that means [7].

The lack of a structured way of working according to information security is obvious in healthcare. There are just not only need for strategies and polices. Also following-up routines and routines for educational programs, evaluation of the strategies and policies etc. are needed.

1.5 Access profile levels

The access profiles are the parts of the access control system where different actors are defined and how they are allowed to process the information. In the existing access control systems there are differences concerning how sophisticated profiles they may create. To maintain a suitable level of information security for patient information, two models are most often combined; the authority model and the logging model.

Depending on the intensity of administration the authority model is required, the most system use the logging model in Swedish healthcare today. According to the SITHS-group, it is important for this kind of method that systems and administration to follow-up the logs exist [18]. There are differences between different care organizations how much resources can be reserved for administration in the models respectively. In most cases, one of the models is given priority to the expense of the other, according to [18]. Some care organizations can have clearly prepared and individually adapted access profiles, which take a lot of time to administrate at the same time as the following-up routines for the log is maintained to a limited extent. Other organizations have very “wide meshed” access profiles instead, which take less time to administrate. Instead, the greater part of the resources is reserved for following-up the logs [18].

Earlier work has shown that if not clear regulations exist and describing how the logs should be managed, there is a main risk that the logs are not checked at all. This implies that the trust for handling of patient information in healthcare can be called in question [10], [29].

Different authors claim that healthcare organizations should emphasize the authority model or “need to know principle” instead of the logging model or “right to know principle”. Gaunt [6] and Smith and Eloff [20], claim that the principal aim is to develop and implement need-to-know authority-controls that would protect patients’ healthcare data. The user would only be allowed to access information necessary to complete his or her job.

Irrespective of what kind of levels of access profiles are used, it is essential to examine the consequences of the choice of levels for the organizations. Since the authority model emphasize the patient’s integrity, the logging model straight forward the need of efficiency of the business and put one’s confidence in the user’s ethics. According to earlier works, mentioned above [19], it is important to find a combination between these two models since the aim is always to reach a balance between the patient’s integrity and the efficiency of the care.

If an authority control should be worked effectively, the trust of the user’s identity must be required. The user must be the person she/he says she/he is. Technically, this can be an easy task to provide when users use own computers at their working places. However, in healthcare there are limited resources also concerning computer equipments and therefore it is common that users use the same computer to perform their tasks. Earlier work has shown that instead of taking time for log-in and log-out procedures, the users use each others’ passwords and perform searches of patient information in someone else’s user’s session [30]. This can lead to crime since secrecy is not covered, or the opposite, authorized information access gives judgments when the logs are checked.

2 Research plan

2.1 The research problem

The problems mentioned above are by no means the only ones in the healthcare area. However, these problems are frequently returned in studies and interviews with involved actors. Both the educational problem and the problem with access profile levels are important issues separately and could be a research problem for research investigation on their own. In my research I will concentrate on further examining security issues related to the integration problem. However, this will not exclude dealing with the other two problems.

The integration problem has been chosen due to the fact that both practice and research has pointed out the necessity of finding safe and sound solutions to the integration of patient information systems within and among the business processes in which they are involved. Information provision in healthcare must be effective such as to contribute to giving patients the care they need. Hence, information security related to business and systems integration is an important issue in particular for maintaining the citizens' trust for healthcare.

In order to come to grips with this problem it needs first of all to be well understood, i.e. we need to make an in-depth study of information security needs including the current state of the art in healthcare processes, in particular where more than one healthcare provider is involved. As a prerequisite to this we need also to find out and analyse what kinds of statutes and regulations that may have an impact on possible solutions. For the research to be useful one needs finally to complement existing business and systems development methodology with useful advice and guidelines for how to reach a satisfactory level of information security.

In accordance with this discussion, a number of research questions may be formulated:

- *What are the information security problems experienced in current heterogeneous healthcare processes?*
- *What are the needs and requirements for information security in healthcare when different healthcare performers integrate their business processes with included patient information?*
- *How do we need to complement business and systems development methodology in order for it to provide adequate support to building solutions with a satisfactory level of information security in healthcare?*

Figure 2 shows in what way the two first questions are related to each other. They are at the same level and are very much connected. When a problem is encountered, it will bring up new needs and requirements, but also when new needs and requirements are discovered, new problems will arise. It is necessary to make a number of iterations in order to give the survey an adequate scope. The results from the first two questions will then be used to build solutions to complement current methodologies.

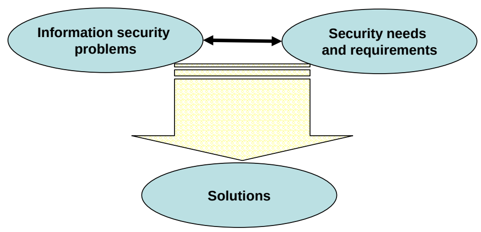


Fig. 2.Research process outline

2.2 Research approach

Some activities in the process can be as follow:

1. A literature survey within the areas of information security, healthcare informatics and healthcare business process integration.
2. Studies of healthcare processes, their need for integration, current integration solutions and the problems caused by these.
3. Observations and interviews with involved actors in the healthcare business processes.
4. Structured interviews with patients involved in the healthcare integrated business processes.
5. Analyze and development work in order to design advice and guidelines as a complement to current business and systems development methodology.
6. Validation process of suggested solution in practice and focus groups with related healthcare actors.

Since I am involved in some on-going research projects in the healthcare area, both in Sweden and Europe, I will use the opportunity to perform healthcare process studies, observations and interviews in co-operation with these projects. In particular, the activities 2 – 3 are conducted within the research projects VITA Nova [27], VITA Nova Hemma [28] and MobiS@ms [13]. However, also other relevant healthcare actors than those engaged in those projects will be involved in the work.

Some work has already been done and below is a brief description of the results and their references.

- The literature survey is an on-going process and will be done in parallel to other work.
- Observations and interviews has been done in subprojects

- MsC dissertation with focus on information security in home healthcare. The result from this work has been published in [29].
- A study about information security in EMR with the user in focus. This study was carried out from a hospital point of view and the result is published in [30].
- Study about system and network security in different heterogeneous healthcare performers. This is a case study in collaboration with the VITA Nova project. The result shall be presented and published at the Security Conference in Las Vegas, March 2005 [31].
- A study at how a process oriented systems architecture can be used in healthcare. The result has been published in the Healthcare Informatics Journal [25].
- An experience report about how a process manager can be introduced in the healthcare business processes. The result has been published in [14].

- The structured interviews are an on-going process in collaboration with the MobiS@ms project and will be finished in June 2005.

The result from the activities 1-4 will form the basis for a PhLic[1] dissertation concerning state of the art and state of practice of information security in integrated business processes in healthcare. The results from the activities 5-6 will then be added to the PhD dissertation.

3 Conclusion

Information security has an important position in the healthcare's information processes. This is not a new thing and plenty of work has been done in this area both in Sweden and other countries. There is still not so much done in the research area about the aspects of information security when different healthcare performers would interoperate their business processes with each other in an efficient way. A research work like this should then be urgent both for the research community and the healthcare domain with respect to their trustworthiness and to find a sufficient level between patient security and patient privacy over the patient care process.

References

1. Blobel, B., (1997) Security Requirements and Solutions in Distributed Electronic Health Records. Computers and Security, 16, pp. 208-209.
2. Blobel, B. and Roger-France, F., (2001). A systematic apporach for analysis and design of secure health information systems. International Journal of Medical Informatics 62, pp. 51-78.

[1] Licenciate of Philosophy – in Sweden a half way graduation to a PhD.

3. Carelink (2003). New agreement for SITHS-CA [on-line]. Available from: http://www.carelink.se [Accessed 3 January 2003].
4. CEN TC 251, prENV 13729: Health Inormatics Secure User Identification - Strong Authentication using Microprocessor Cards (SEC-ID/CARDS), (1999).
5. Dahlin, B. and Arnesjö, B. (1996). Datorjournalen. In: G. Petersson and m. Rydmark, editors Medicinsk Informatik. Liber Utbildning, chapter 6 (in Swedish).
6. Gaunt, N. (1998). Installing an appropriate information security policy. Internation Journal of Medical Informatics, 49, pp. 131-134.
7. Gaunt, N. (2000). Practical approaches to create a security culture. International Journal of Medical Informatics, 60, pp. 151-157.
8. Gritzalis, D. (1997) A baseline security policy for distributed healthcare information system. Computer & Security Vol 16, No 8, pp. 709-719.
9. SIS (2003). SIS Handbok 550. Terminologi för Informationssäkerhet. Stockholm 2003 (in Swedish).
10. Karlsson, K., (2003). Right-to-know inom hälso och sjukvården. Msc thesis, Department of Computer and Systems Sciences. Stockholm, Stockholm University & Royal Institute of Technology. Available from: http://www.simovits.com/arhcive/CatKarl.pdf [Accessed 2 April 2003] (in Swedish).
11. Katsikas, S. K. (2000). Healthcare management and information system security: awarnesss, training or education? International Journal of Medical Informatics, 60, pp.: 129-135.
12. Louwerse, K. (1998). Availability of health data; reguirements and solutions Chairpersons' introduction. International of Medical Informatics, 49, pp. 9-11.
13. MobiS@ms (2003). Projektplan Mobis@ms [online] http://www.lime.ki.se/mobisams. [Accessed Jan 2005] (in Swedish).
14. Perjons, E., Wangler, B., Wäyrynen, J. and Åhlfeldt, R-M. (2004). Introducing a Process Manager in Healthcare: An experience report. In Proceedings of the Ninth International Symposium on Health Information Management Research (iSHiMR), Sheffield, June 15-17, 2004, pp. 71-86.
15. Poulymenopoulou, M., Malamateniou, F. and Vassilacopoulos, G., (2003) Specifying Workflow Process Requirements for an Emergency Medical Service. Journal of Medical Systems, 27(4), pp. 325-335.
16. SEMA (1999). FA22. Swedish Emergency Management Agency. Växjö, Grafiska Punkten. 3 (in Swedish).
17. SITHS-projekt, (1999). Informationssäkerhet i vårdprocessen: Krav beskrivna i generella användningsfall utifrån vårdcenarion, Stockholm: Säker IT i Hälso- och sjukvården, ISBN 91-7188-565-X (in Swedish).
18. SITHS-projekt, (2000a). Tjänster för att uppnå informationssäkerhet i hälso- och sjukvården. Stockholm: Säker IT i Hälso och Sjukvården, ISBN 92-7188-607-9 (in Swedish).
19. SITHS-projekt, (2000b). Infrastruktur för informationssäkerhet i hälso- och sjukvården. Stockholm: Säker IT i Hälso- och Sjukvården, ISBN 92- (in Swedish).
20. Smith, E. and Eloff, J. H. P. (1999). "Security in healthcare information systems - current trends." Internation Journal of Medical Informatics 54: 39-54.
21. Spri (1996). Behörighet, säkerhet och sekretess - krav på system med patientinformation. Spri rapport 419. Stockholm, Spri (in Swedish).
22. SWEDAC (2003). SWEDAC:s föreskrifter STAFS 2003:13. [on-line] http://www.swedac.se [Accessed May 2004] (in Swedish).
23. Sågänger, J. and Utbult, M. (1998). Vårdkedjan och informationstekniken, Teldok. (in Swedish)

24 Teldok (2004). Patientdata - brist och överflöd i vården. Teldok rapport nr 154. ISSN 0281 - 8574 (in Swedish).
25. Wangler, B., Åhlfeldt, R. and Perjons, E., (2003) Process Oriented Information Systems Architectures in Healthcare. *Health Informatics Journal*, Vol 9(4), 253 - 265, December 2003
26. Wenzlaff, P., Blobel, B. and Pharow, P., (1999) Health Professinal Cards: awarenss for security in Health Care Networks, in: F. Sicurello (Ed.), Healthcards '99, XASI, Milan, 1999, pp 121-125.
27. VITA Nova, *(*2002). Projektplan VITA Nova I [online]. http://www.ida.his.se/ida/research/vitanova [Accessed Jan 2005] (in Swedish).
28. VITA Nova Hemma, *(*2003). Projektplan VITA Nova Hemma, [online]. http://www.ida.his.se/ida/research/vitanova [Accessed Jan 2005] (in Swedish).
29. Åhlfeldt, R.-M. (2002). Information Security in Home Healthcare; A case study. In the Conference Proceedings of the Third International Conference of the Australian Institute of Computer Ethics (AiCE) 2002. Sydney, Australia, 30 September 2002, pp. 1-10. Eds. M. Warren and J. Barlow. Australian Institute of Computer Ethics.
30. Åhlfeldt, R-M. and Ask, L. (2004). Information Security in Electronic Medical Records: A case study with the user in focus. In Proceedings of the 2004 Information Resources Management Association International Conference. New Orleans, USA, May, pp. 345-347.
31. Åhlfeldt, R-M. and Nohlberg, M. (2005). System and Network Security in a Heterogenous Healthcare Domain: A Case Study. In Proceedings of the 2005 Security Conference, Las Vegas, USA, Mars, to be appered.

Probabilistic Time Management for Automated Business Processes

Horst Pichler (PhD student), Johann Eder (PhD supervisor)
University of Klagenfurt, Institute for Informatics-Systems, Austria
horst.pichler@uni-klu.ac.at, johann.eder@uni-klu.ac.at

***Abstract*—Empirically learned characteristics of a business process, like branching probabilities and average execution durations of activities, are usually used for process simulation and the creation of scheduling scenarios as a basis for process optimization. We utilize this information by calculating a probabilistic timed process model based on the process structure, time properties and probabilistic information. It yields possible execution intervals for activities in order to meet defined time constraints, where the probabilistic information provides information about the likelihood of a future deadline violation. Another application area is the calculation of the probability for expected (remaining) process execution times. In order to cope with the interoperability aspect of business processes we currently examine the possibilities to map these workflow-based concepts on composite web service (CWS) environments. Slow autonomous web services, invoked by a CWS, can have an disastrous impact on the overall process response time. Thus techniques are needed predict the process duration based on the anticipated response time of participating web services, which enables us too avoid these services or to optimize them for faster execution.**

I. Time Management

Typically, time violations increase the cost of business processes because they require some type of exception handling. Therefore, the comprehensive treatment of time and time constraints is crucial in designing and managing business processes. Process managers need tools that help them anticipate time problems, pro-actively avoid time constraints violations, and make decisions about the relative process priorities and timing constraints when significant or unexpected delays occur. These are established problems in workflow management, but most of the provided solutions suffer from the uncertainty of time information in processes. This vagueness stems mainly from two aspects: The duration of a task can vary greatly and in a workflow different paths may be chosen with decisions taking place during the execution. Some approaches try to address this problem by introducing time intervals (e.g. best and worst case), or suppose some kind of distribution (e.g. normal distribution) of duration values. The probabilistic approach, presented in this paper, offers a new solution to these shortcomings and enables more precise predictions about the likelihood of the future behavior of a process execution.

II. Time histograms

Activities as part of workflows are capable of hiding complex business processes with greatly varying durations. Thus simple duration representations, as mentioned above, fall short because they are to imprecise. They all do not take into account that the existence of conditional structures in the control flow may result in a non-evenly distributed duration with multiple peaks. Thus the concept of time histograms

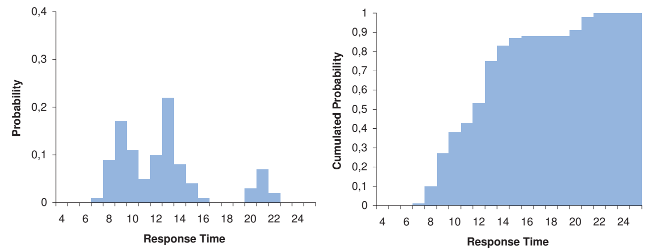


Fig. 1. Response time histogram

(THs) was introduced in [1][2] as structure to represent a distribution function on a time property (e.g. activity duration or process response time). Figure 1 shows an example of the graphical representation of a TH representing the response time of a process. It (on the left hand side) tells us that that the bigger part of future executions will last between 7 and 16 time units with peaks at 9 and 13 plus some additional outliers around 21. The time unit depends on the chosen minimum granularity, e.g. millis, seconds or minutes. According to the associated cumulated representation (on the right hand side) we can for example state that there is a likelihood of 88% that the activity will last 16 time units or less, thus we are enabled to determine the probability for a given threshold duration or vice versa. THs are either generated by expert estimations, the extraction of empirical information from process execution logs or calculated, based on the the structure of an underlying business process. THs are also used to calculate execution intervals for each activity in the process, bounded by earliest possible start times and latest allowed end times (in order to hold process deadlines) for each activity in the process. This information may be used for time constraint validation during build time or pro-active task scheduling during run-time.

III. Workflow graph and duration histogram calculation

[1][2] show how different types of time histograms are calculated, based on workflow definitions using sequences, conditional routing (and-splits/joins), parallel routing (x-or-splits/joins) or iterations (loops). As prerequisite for the calculation we define a (directed acyclic) workflow graph holding activities and the control flow dependencies between these activities. Additionally probabilistic information about the branching behavior of a process and the duration of its activities, represented as THs, must be provided. Until now we allowed only well-formed workflow graphs. A well-formed graph has to adhere to certain constraints: a) For every split-node there exists exactly one join-node of the same type, e.g.

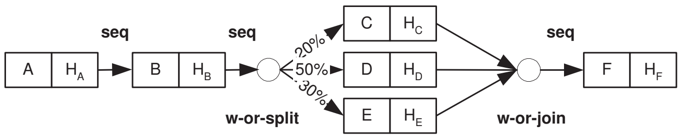


Fig. 2. Well-formed workflow graph

an or-split must have an associated or-join, b) every path that originates from a split-node must merge in the associated join-node, and c) dependencies originating from activities between split/join-pairs must not leave their paths or connect to activities outside of these pairs. Figure 2) shows a simple graph containing activities (A,B,C, etc.), probabilistic duration information (THs named H_A,H_B,H_C, etc.) and two different types of control flow structures: sequence and weighted x-or, which demands that between split an join exactly one out of many paths must be executed, depending on a given branching condition or choice. The weight represents the expected branching probability of each path. To calculate for instance the TH for the workflows response time it is necessary to traverse the graph from the first to last activity while summing up and weighting the activity-DHs according to control structures and branching probabilities. The result of the calculation is a DH for the workflow consisting of tuples for every possible overall duration with its according probability. For further details see [1][2].

IV. Mapping the concepts on composite web service environments

The next step in the evolution of business process automation are composite web services (CWSs) to support business processes within organizations as well as business processes spanning several organizations like supply chains. Thus the most critical need in companies will be to provide services with a better quality than their competitors. To assess the quality of service (QoS) it is necessary to define measures which are significant indicators for certain quality aspects, where e.g. expected or guaranteed process duration ranks among the most important dimensions. Slow autonomous web services, invoked by a CWS, can have an disastrous impact on the overall process response time. Thus techniques are needed predict the process duration based on the anticipated response time of participating web services, which enables us too avoid these services or to optimize them for faster execution. Due to the resemblance between the concepts workflow and CWSs, an adoption of the above presented seems reasonable. The following subsections outlines our current and future research projects and arising problems for a solution using executable processes in BPEL4WS.

A. Response time histograms

Again the response time of a process is either detected empirically or calculated. Workflow activities are mapped to invoked web services and the duration is mapped to response times of these services. The histogram concept will prove valuable as infrequent response time outliers, caused by delays stemming from communication problems, will be located in the upper regions of the cumulated histogram. This enables us to cut them of by defining a threshold less than 100%. E.g. the response time of a service with a likelihood of 98% will not include very infrequent delays but still yield acceptable forecasts.

B. Adapting the workflow graph for BPEL4WS definitions

The approach explained above was designed for graphs representing well-formed workflow definitions. These constraints are too rigid for possible compositions defined with different types of primitive activities (invoke, receive, reply, etc.) and structured activities (sequence, link, flow, etc.) available in BPEL4WS, which may result in non-well-formed workflows. Therefore an extended graph-representation which allows non-well-formed structures must be defined and the calculation algorithms must be adapted. Future research will build upon pattern based analysis of BPEL4WS. Another research topic will be the implementation of a parser to extract a non-well-formed graph from a BPEL-definition.

C. Gathering data for response time histograms

In workflow systems empirical data can easily be extracted from the workflow log. In BPEL-environments this can either be done by the process-execution-engine or, if the engine has no logging-feature, be implemented as part of the process itself. However, to generate response time histograms and branching probabilities it is necessary to log the system times of each request and each response as well as the control flow of all instances. This issue tends to be a major problem, especially in flexible environments where autonomous web services, accessed by the composition, are frequently changing. For these cases web service response times could be stored and administered by a trusted third party, which offers an interface to access response time statistics, similar to or as an extension of a Web Service Level Aggreement-architecture.

D. Possible application scenario

Equipped with the prerequisites explained above a possible application scenario consists of: a) Web service compositions published as web services to the outside world. b) These web services will have additional interfaces yielding the expected service response time as TH. c) The TH can either be generated by accessing statistical data or calculated by applying the above mentioned algorithms on the structure of the composition. The services invoked by the composition are themselves equipped with these interfaces. d) Dependent on the resulting TH the requestor is enabled to make several decisions (like using this service or selecting an alternative one or for providers: optimizing the process for faster execution, etc.).

References

[1] J. Eder, H. Pichler, W. Gruber, M. Ninaus. *Personal Schedules for Workflow Systems*. In Proceedings of the International Conference on Business Process Management, Eindhoven, The Netherlands, Springer Verlag (LNCS 2678), 2003.

[2] J. Eder and H. Pichler. Duration Histograms for Workflow Systems. In Proceedings of the Working Conference on Engineering Information Systems in the Internet Context, 2002, Kanazawa, Japan, Kluwer Academic Publishers, page 239-253.

Knowledge Management for Interoperable Quality Management System

Francesca Scorziello[1]

[1] Politecnico di Bari, Viale Japigia, 182,
70126, Bari Italy
scorziellof@poliba.it

Extended abstract

Quality Management System (QMS) can be defined, according to the international norms as “that part of organisation’s management system that focuses on the achievement of results, in relation to the quality objectives, to satisfy the needs, expectations and requirements of interested parties, as appropriate”. [4]

Having in mind that a system is a set of heterogeneous entities interacting for achieve a common goal, also a QMS can be seen as a set of some interacting subsystems (for ex. management subsystem, operative subsystem, control subsystem and support subsystem). The first one allows the organisation definition of policies and targets, for the whole organisation, planning of each activities, including resources for their execution and identification of procedures, ruling flows of operations and also of guidelines for Quality Manual. The operative subsystem generally consists of design area, industrialisation area, production area, testing area and supplier area. It allows the operative management of an enterprise, by the definition of operative processes. The control subsystem allows examination of the organization, as concerning quality targets, designing and execution of periodic audit and analysis of their results. The last subsystem consists of each activities allowing the efficient working of the other subsystems, for example training, prevention and comparison with external enterprises.

Not all organisations, implementing a QMS, identify the previous subsystems, someone prefer having management, operative and support subsystem. That implies a different point of view about how managing and deploying activities.

Let us assume that an organization have to collaborate with other organizations to achieve a common goal; in this case it is necessary to take into account a new paradigm in implementing the respective QMS’s: interoperability. In general term, it consists in assuring the ability of parts of the QMS to support cooperation, under quality assurance constraints, with one or more other parts of the cooperating organisations. The problem turns then to be how to achieve an efficient and effective interoperation to pursue a common goal within a defined interval of time while addressing quality targets [2], taking into account that the identification of which part of the two organizations can interoperate could be not easy. In fact, according to the previous considerations, the cooperating organisations not necessarily have identified the same subsystems. Design criteria for an Interoperable Quality Management System (IQMS) need to be appropriately defined to support quality targets in the cooperating processes; taking into account that the cooperating organizations, implementing a IQMS, needs to restructure the management approach as well as the target of performances. They have also to identify operative praxis, responsibility and tasks for each interoperating process, as well as the abilities and knowledge necessary for each involved activity. Designing a IQMS is also an occasion for creating shared experiences (building blocks of organizational memory): the working together of organization members creates shared understanding as well as common memories [5], and ISO documentation houses those memories for future workers. In addition, as a codebook, ISO 9000 provides a process of knowledge codification.[1] All these characteristics are essential for knowledge creation practices, then some basic principles of knowledge management have been recalled herewith to better structuring the cooperative processes, during the design of an IQMS. New modelling methodologies for supporting the design of an IQMS have also been applied, taking into account the impact of the design oriented to quality for the organisation. According to this, advanced enterprise models has been developed to support the simulation of design effects, the analysis of their performance and the influence on management of their operations.[3]. Enterprise models have been structured for the implementation of the necessary interoperability features by including interface descriptions and infrastructure characteristics. A case study to explain evidences related to interoperable quality management system is also provided.

References

1. Bénézech, D., Lambert, G., Lanoux, B., Lerch, C., Loos-Baroin, J. :Completion of knowledge codification: an illustration through the ISO 9000 standards implementation process. Research Policy, Vol.30, No. 9, pp.1395-1407 (2001)
2. Dassisti M., Scorziello F, Enterprise model interoperability: a case study of two university laboratories, SEMINARIO CIRP '04 SORRENTO 29 GIUNGO-2 LUGLIO, (2004)
3. Fox Mark S., Gruninger Michael, , Enterprise Modeling, AI Magazine, pp. 109-121 (1998)
4. ISO Quality Management Systems–Fundamentals and Vocabulary2001, ISO, 2nd ed., Switzerland. (2000)
5. Zuckerman, A.: Gaining added value from the year 2000 ISO 9000 revisions. World Trade, Vol.13, No. 2, pp. 60-62 (2000)

Interoperability problems: Management of evolution of collaborative enterprises

Séverine Blanc

LAPS/GRAI – University Bordeaux 1 / ENSEIRB / UMR CNRS 5131
351 cours de la Libération – 33405 Talence cedex – France
Tel: +33 5 4000 2409 – Fax: +33 5 4000 6644
E-mail: severine.blanc@laps.u-bordeaux1.fr

Abstract. The actual economic context forces the enterprises to collaborate together to survive against an increasingly aggressive competition. This paper starts with several definitions. Then, it presents a method to solve some enterprise modelling interoperability problems. This method is focalizing on the supply chain context and includes four parts about evolution management of one or several enterprises, performance of the supply chain, how to model only the flows that we need in this perspective and how to take into account the human aspects.

1 Introduction

In the current strong competitiveness industrial context, enterprises must react quickly to the market changes. In order to face this problem, enterprises must collaborate together. This implies at one hand high communication between their information systems and at the second hand the compatibility of their practices. An important work of change must be done for the practices standardization and harmonization. This is the concept of Interoperability.

This paper aims to present the basic definitions and concepts of interoperability and the research developed currently within the LAPS GRAI group at University Bordeaux 1: Management of evolution toward interoperable collaborative Enterprises using concepts of enterprise modelling.

2 Definitions and research domains on Interoperability

2.1 Definitions

Before presenting this work, a necessary set of definitions is given, concerning Interoperability, Collaborative Enterprises and Supply Chain to understand the topic.

Generally, Interoperability is defined as the ability of two or more systems or components to exchange information and to use the information that has been exchanged[1 2]. More precisely, IDEAS project defines interoperability as the ability of interaction between enterprise software applications. The interoperability is considered achieved if interactions can, at least, take place at three levels: data, application and business process with the semantics defined in a business context[3]. This definition is supported by the IDEAS interoperability framework as showed in Figure 1.

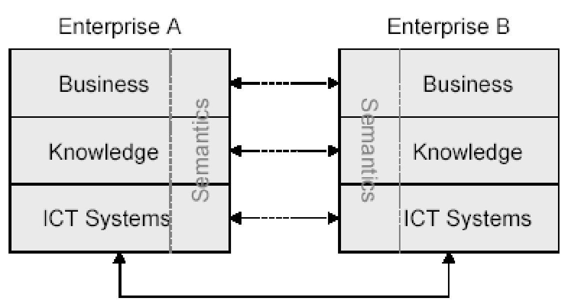


Figure 1. IDEAS interoperability framework

To develop interoperability in Collaborative Enterprises, which is defined as an organization to allow enterprises to collaborate on numerous common projects during a long time, the supply chain context was chosen. The supply chain includes all the activities associated to the transformation and circulation of goods and services, since the extraction of the raw materials to the final customer[4].

2.2 Research domains on Interoperability

According to the standard ISO, Enterprise Modelling (EM) is defined as the act of developing an enterprise model which is a representation of what an enterprise intends to accomplish and how it operates.
More precisely, EM is the representation of the structure, the behavior and the organization of the enterprise according to different points of views:

- Functional, Informational, Physical (Business), Decisional, Processes…
- Technical, Economical, Social, Human…

with two interconnected visions:

- Global: System Theory: The global view of the enterprise: objectives, structure, functions, evolution of the enterprise (dynamic), links with the environment;
- Local: Detailed description according to the concepts of activities and processes.

The role of enterprise modeling is to represent, understand and analyze through an enterprise model, the running of an enterprise in order to improve its performances[3].

The role of EM in Interoperability is to define interoperability requirements and to support solution implementation. This contributes to solve interoperability problems by increasing the shared understanding of the enterprise structure and behavior. Enterprise Modelling provides methodologies for the identification of connected roles, objects and processes between enterprises from different perspectives.

Several problems are related to Interoperability and EM:

- The enterprise models of both parts are not exchangeable (i.e. built using two different languages)
- The same term used by two parts does not mean the same thing (semantic problem)
- The models of both enterprises show differences in practices which are not aligned (output of the first does not correspond to input of the second)
- The model of both information technology (IT) systems shows incompatibility in information exchange…

Enterprise Modelling has a significant role in Interoperability, particularly in term of analysis to target the problems which can appear in an approach of implementation of Interoperability and to solve these problems.

3 Researches for Interoperability

The problematic of our research work is how to allow the interoperability between several enterprises in the frame of a supply chain on the basis of enterprises which don't collaborate.
To solve the interoperability problems, we can gather them in three distinct parts: problems involved in the ontology, those related to architecture and platform and those related to model of enterprise. The first allows having a common vocabulary, the second allows the interoperability by the technical aspects (software, hardware, net…) and the third models the supply chain to allow having interoperable practices at the interfaces. To solve the problems related to ontology, we have two options: either we set up a common and global ontology in all enterprises of the supply chain but the implementation is difficult and tiresome, or we set up a common ontology only to the boundaries of the enterprises. The problems related to architecture and platform aren't in our domain, therefore, we consider that they are solved. To solve

the problems related to the model of enterprise, we use the GRAI Methodology. Our start point is enterprises which don't contribute between them and we want to bring them toward the interoperability. We start from the GRAI Evolution Method (GEM[5]) to apply this work in the case of collaborative enterprise.
Therefore, to bring an answer to some of the enterprise modelling interoperability problems, we need to develop a method which has the following functionalities:

- To manage the evolution of enterprise with the definition of different steps;
- To manage the performance of the supply chain in this entirety. The notion of performance is very important because it allows to bring the activity and to share the information, to promote the cooperation between the function in the enterprises and between the members of the supply chain, and the will to increase the vision angle inside the supply chain;
- To model only the information, the flows and the services which concern interoperability of the supply chain. We don't speak about boundaries of an enterprise toward another enterprise but we speak about boundaries of the supply chain. Indeed, two enterprises don't need to be completely interoperable, but they need to be interoperable at the interface. For this reason, we have to define a supply chain boundary to separate the services which collaborate from the others;
- To take into account the human aspect i.e. the communication between different people and the human psychological aspects in the evolution of their enterprise. Indeed, in the evolution management, people are often recalcitrant to change.

3.1 Evolution management of enterprises

This approach is based on the research work of N. Malhéné[5]: The GEM approach (see Figure 2). It describes evolution process management of industrial systems. In the frame of interoperability, it is proposed to use this method, developed to manage the evolution management of an isolated enterprise, to manage the combined evolution of several enterprises to obtain an interoperable supply chain. The principle of this approach is to build system evolution like continuous processes. In practice, the evolution process is made of a sequence of steps representing the evolution of the system status.
The AS IS represents the model of existing system. The components of the system are here described and formalised: it is possible to understand better how the system is running and also to detect the points to improve.
The SHOULD BE corresponds to the strategic objectives of the system. In our approach it is described through the Business Plan of the enterprise.

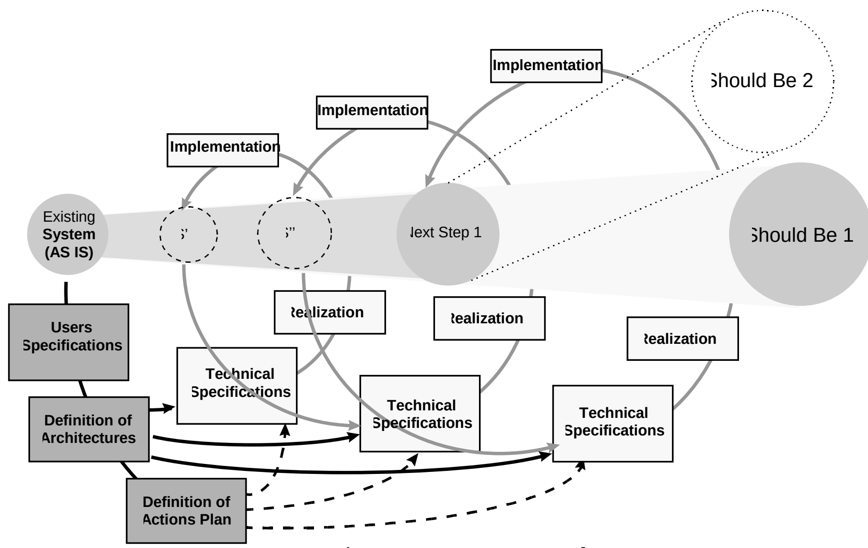


Figure 2. GEM Approach

The NEXT STEP is an intermediate stage between the AS IS and the SHOULD BE. It corresponds to the future system which will be implemented.
The USER SPECIFICATIONS correspond to the comparison between the NEXT STEP and the AS IS models. From this, the TECHNICAL SPECIFICATIONS, which include the Organisation, the Information Technology and eventually, the Physical part, are deduced. The Action Plan determines several evolution projects with a limited duration and investment.
To complete this approach, the human aspects will be taken into account. Indeed, these aspects are often those which bring problems during the evolution.
To validate each step, performance indicators are established.

3.2 Evolution toward Interoperability

There are a lot of reasons of measuring performance of the supply chain. First, it is necessary to determine the interactions between the performance of each actor and the one of the chain as a whole. Second, it is required to share the positive and negative effects of evolution into the supply chain. Third, it is necessary to align the activities and to share the information.

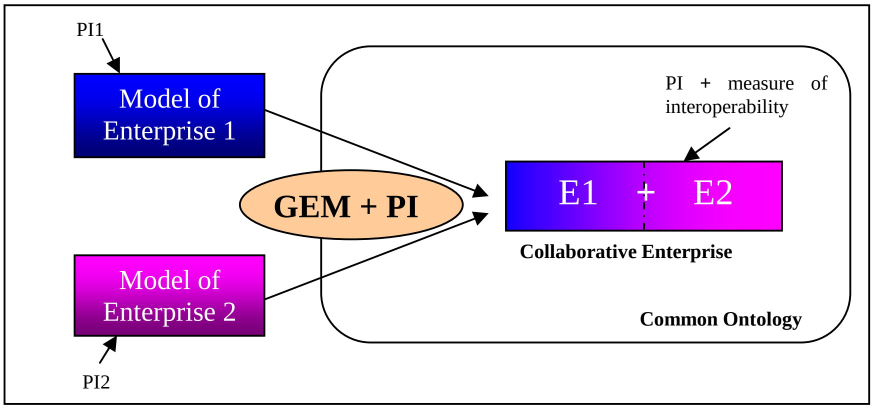


Figure 3. Approach of evolution management

In our approach (Figure 3), performance indicators (PI) are defined in the enterprise before the beginning of evolution. These first indicators will be used to set up a favourable climate to the change in order to prepare the enterprise according to the three following aspects:

- Human;
- Material;
- Knowledge management.

Then, from the beginning of evolution, other performance indicators are established to validate each step of the method. These information will allow to determine the new "Should Be".

Finally, after having obtained the collaboration into the supply chain, final performance indicators are defined in order to manage and control the supply chain in its entirety.

3.3 Modelling of enterprise operational processes

The objective of this model is to describe in the same formalism the control flows, the material flows and the information flows making up the operational processes of an enterprise. It combines some principles which come from algebra of processes[6] for the functional and behavioural part and from statecharts for the notion of state and reactivity of system by sending events and messages. It allows to describe as well the transformational processing as the event processing in a unified formalism. The central concept is the processes notion. It defined many concepts, only the main ones are described in this paper.

A *processes* is defined as a partially ordered set of step forming a data processing sequence delimited by a beginning (START) and an end (FINISH). Formally, one can specify processes P by a 5-uplet:

$$P = \langle Pid, \alpha P, \beta P, \delta P, SP \rangle$$

A step of processes is either under-processes, or an elementary step called *activity*. It is defined by behaviour rules of the form:

WHEN (condition) DO (action)

The processes are started by events. An event represents a change of state in the observed system. It appears at a given moment when a condition becomes checked. It can be carrying information in the form of a object view. One defines it by a 4-uplet form:

$$E = \langle Eid, q, OV, \tau \rangle$$

The *object view* defines an enterprise object state at a given moment. We can distinguish two types of object view: physical view describing the physical object (material flow) and the information view describing an object with the data-processing direction (information flow).

The activity is an elementary stage in processes. One defines it by a 10-uplet form:

$$A = \langle Aid, FIA, FOA, CIA, COA, RIA, ROA, \delta A, IA, SA \rangle$$

The *functional entities* represent the active resources (i.e. the agents or actors) of the considered enterprise. They provide and carry out the necessary functional operations by the activities. We can distinguish three fundamental types of functional entities in an enterprise: the men, the machines (built material entities able to carry out operations on request) and the applications (systems software). In a formal and generic way, a functional entity is defined by:

$$R = \langle Rid, OVR, CapR, FOR, fR \rangle$$

3.4 Weak Aliens Problem (WAP)

We can differentiate two different cases of heterogeneity: material heterogeneity and Semantic Heterogeneity.

In the case of material heterogeneity, for a long time, the distributed systems have been few interoperable. Today, large progress have been done and architectures of multi-agents systems in particular have strongly contribute to reduce the question of information transport, in an independent way of the material conditions i.e. machines and/or subjacent operating systems.

In the case of semantic heterogeneity, we can say that the applications which interact on the basis of agent have often been defined and built by different people, in different places at different times, with different aims, with different vocabularies... So the agents live in a "semantically open" world[7].

According to J-P Sansonnet[7], three different situations of communication exist:

- NAP: Global ontology which hold all particular ontologies and there is no problem of semantic heterogeneity;
- SAP: The agents do not have any intersection and they can't communicate;
- WAP: Ontologies are different but however they have an intersection.

Now, we wonder if we have to eradicate or confront the heterogeneity.

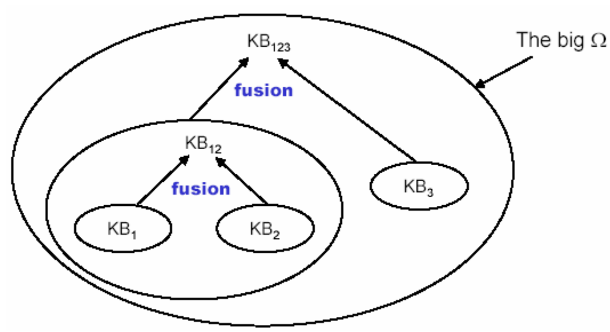


Figure 4. Eradication of heterogeneity

The semantic heterogeneity by the fusion of all particular ontologies in a global ontology can be eradicated (see Figure 4).

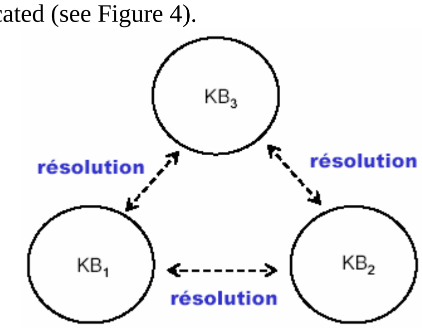


Figure 5. Confrontation of semantic heterogeneity

The semantic heterogeneity can be confronted by the resolution of the individual problems. In this perspective, three strategies can be defined to confront the heterogeneity. Firstly, the concepts can be expressed within the meaning of description logics (see Figure 6)

Kbase Agent1	**Kbase Agent2**
Termes : A,B,C, D, E TBox : $A \equiv E \wedge C$ $D \equiv \forall r.C$ $B \leq E$ $A \leq C$	Termes : A,B,C, E, F,G TBox : $F \equiv \forall s.C$ $G \leq C$ $E \leq G \wedge F$ $A \equiv B \wedge C$ $A \leq C$

Figure 6. Description logic

Secondly, they can be expressed under the form of lexical concepts graph (see Figure 7)

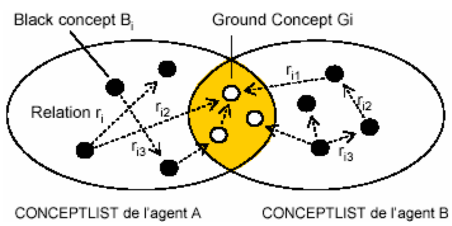


Figure 7. Graph of relations

Thirdly, we can express them under the form of concepts expressed in algebraic topology (see Figure 8).

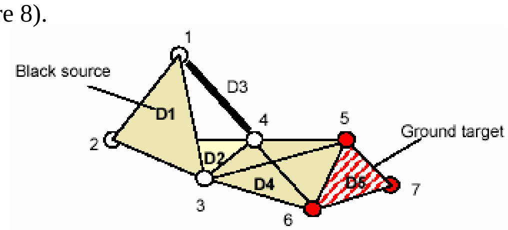


Figure 8. Simplicial complex

To apply this last representation on a more concrete example in our domain, we can use this graphs to represent the decision frame (blue arrows (Figure 9) of a GRAI grid[1] example (Figure 10).

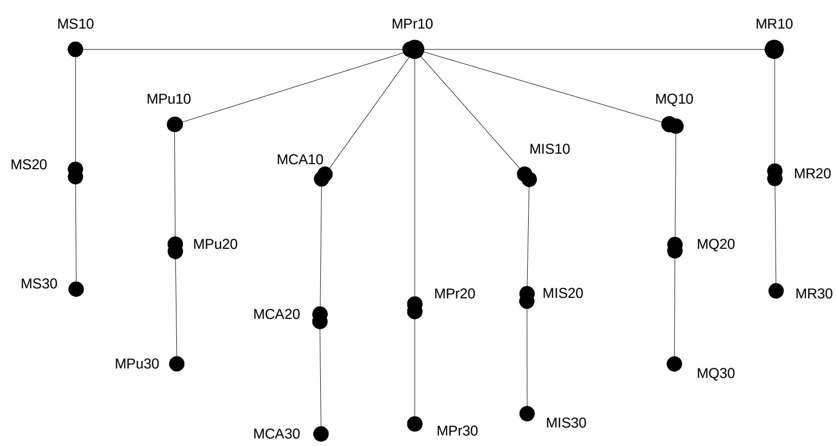


Figure 9. Simplicial complex representation apply to the GRAI grid example

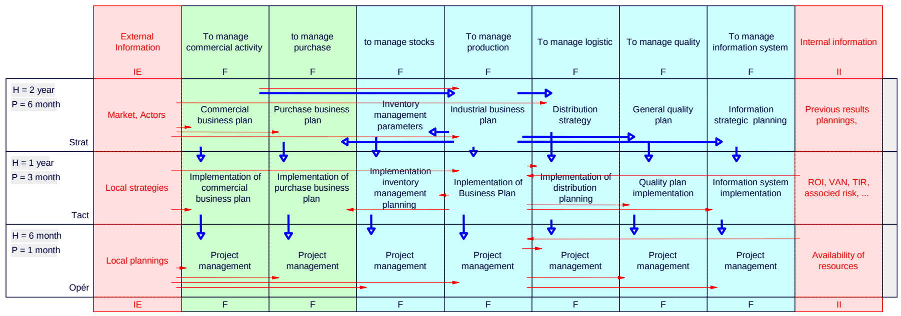


Figure 10. The GRAI grid example

4 Conclusion

This paper has presented a method to manage the evolution of enterprises from heterogeneity to interoperability. This method is articulated around four parts, each one filling a particular task:

- To manage the evolution of enterprise;
- To manage the performance of the supply chain in its entirety;
- To model only the information, the flows and the services which concerned interoperability of the supply chain;
- To take into account the human aspect.

These research works are only beginning. In the future, we will detail each aspect to develop the global method.

5 References

1 ATHENA (2004), Advanced Technologies for Interoperability of Heterogeneous Enterprises Networks and their Applications, FP6-2002-IST-1, Integrated project Description of Work, 2004.

2 INTEROP (2003), Interoperability Research for Networked Enterprises Applications and Software, Network of Excellence, Annex 1-Description of Work, 2003.

3 IDEAS project deliverable (WP1-WP7), Public Reports, www.ideas-roadmap.net, 2003.

4 P. Hauguel, E. Viardot, De la supply chain au réseau industriel, L'expension Management Review, p 94-100, June 2001.

5 GEM, N. Malhéné, Gestion du processus d'évolution des systèmes industriels – Conduite et méthode, thesis of university Bordeaux 1, January 2000.

6. Hoare C.A.R.E. *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs, 1985.

7. Sansonnet, J-P.: Approches de l'hétérogénéité sémantique entre agent informationnels http://www.limsi.fr/Individu/jps/research/weakaliens/weakaliens.htm, 2004.

Inter-Organisational Collaboration on the Process Layer

Christina Tsagkani

National & Kapodistrian University of Athens,
Dept. of Informatics and Telecommunications
Panepistimiopolis, Ilissia, Athens 15784, Greece
tsagkani@di.uoa.gr

Abstract. Today the competitive marketplaces require that the enterprises should be more flexible, innovative and responsive to their needs. Therefore, the enterprises and specially the small and medium sized ones, in order to gain a competitive advantage, should get rid of their traditional business models and adapt new ones to facilitate collaboration. This paper is a research plan aiming at a PhD degree which concentrates on dynamic collaboration between companies and in particular on the business process layer of the business cooperation framework.

1 Introduction

Nowadays organisations are moving or have already moved their main operations to the Web, to take advantage of the potential of more automation, efficient business processes, and global visibility. They form collaboration networks that value speed, quality and have the ability to react dynamically according to individual objectives (e.g. produce a product, provide services, outsourcing). These collaboration networks have attracted the interest of research communities but there are still many issues that need to be considered.

This paper is a research plan aiming at a PhD degree which concentrates on dynamic collaboration between companies and in particular on the business process layer of the business cooperation framework. It is thus organised as follows. In Section 2 there is an introduction on the business processes and business process management. Different types of interoperability and forms of collaboration are presented next in Section 3. In the same section, the different levels of cooperation framework are described and some research issues that arise when different companies want to interoperate are stated. In Sections 4 and Section 5 the Workflow Management Systems and Virtual Enterprises are briefly described and some of the corresponding problems are mentioned accordingly. Finally Section 6 depicts the research plan and outlines future steps that are going to be followed.

2 Business Processes and Business Process Management

Business processes are market-centred descriptions of an organisation's activities. That is, business processes are collections of activities that support critical organisational functions in realizing an objective or policy goal.

Depending on the type of participants, business processes can be classified in:

- Public processes (external), that are processes performed in collaboration with others outside the borders of the organization (e.g. suppliers in a supply chain scenario, partners in a virtual enterprise scenario and customers in a B2B scenario).
- Private processes (internal) that are performed and acknowledged only within the enterprise.

Business Processes need to be effectively managed in order to fulfil their objectives.

Business Process Management (BPM) can be defined as follows: *Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information* [12]. It should be noted that this definition restricts BPM to operational processes that include processes at the strategic level. This means, processes that cannot be made explicit are excluded. It is very important that systems supporting BPM need to be "process aware", i.e., without information about the operational processes at hand little support is possible.

The enterprises that advocate Business Process Management, fully control their implicit processes and they are usually aided by a universal process language like BPML, BPMN and BPEL [12] that enables partners to execute on shared vision, in order to interoperate with other enterprises, to communicate and understand each others processes in detail, to jointly design processes and manage the entire lifecycle of their business improvements initiatives.

BPM, in order to be effectively applied, needs to be supported by a Business Process Management System (BPMS). A BPMS enables companies to model, deploy, and manage mission-critical business processes that span multiple enterprise applications, corporate departments, and business partners that may be behind the firewall and over the internet. Examples of BPMS include: Tibco Staffware Process Suite [20], WebMethods Business Process Management [21], Vitria BusinessWare [22], etc. The BPMS is a new category of software and opens a new era of IT Infrastructure [1], [2].

3 Environmental Influences and Business Collaboration

A major part of the global economy consists of small and medium-sized enterprises. These enterprises, in order to stay competitive and survive in a rapidly changing environment, need to be part of collaborative networks of enterprises [3].Thus they need to focus on core competencies and mission-critical operations and outsource everything else.

Collaboration is defined in [9] as the agreement among a set of participants (e.g. Companies, Web services) to achieve a common goal or specified outcome in a shared process. Besides, interoperability is the ability of a system or a process to use information and/or functionality of another system or process by adhering to common standards [10].

As soon as different enterprises are able to interoperate or collaborate, they have been empowered with effective capability of mutual communication of information, proposals and commitments, requests and results. In general, interoperability covers technical, semantic and pragmatic interoperability [4]:

- Technical interoperability means that messages can be transported from one application to another.
- Semantic interoperability means that the message content becomes understood in the same way by the senders and the receivers. This may require transformations of information representation or messaging sequences.
- Finally, the pragmatic interoperability captures the willingness of partners for the actions necessary for the collaboration. The willingness to participate involves both capability of performing a requested action, and policies dictating whether the potential action is preferable for the enterprise to be involved in collaboration.

The collaboration between enterprises may have different forms:

- Collaboration based on specific standards e.g. EDI. This is a unified approach to collaboration.
- Collaboration by following an integration approach, i.e., processing platform integration, data integration, etc.
- Collaboration by joining a virtual enterprise which is usually dynamic. This means that the partners are loosely coupled and form short term partnerships. In this form of interoperability a federated approach to collaboration should be applied, that establishes and maintains collaboration between autonomous local services, each of which runs a local business process [4].

Business cooperation means that a network of companies is working together in order to deliver products and services that no-single company is able to deliver itself. This cooperation is multi-facet and should be examined regarding the following levels [5]:

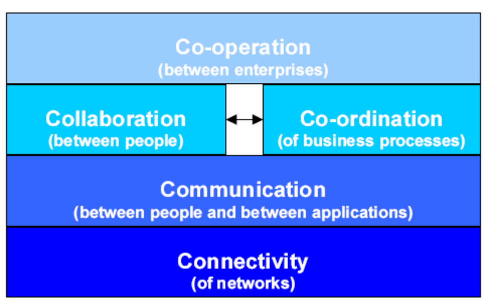


Fig. 1. Levels of Business Cooperation [5]

- The connectivity level, that uses protocols (e.g TCP/IP) in order for the applications of different enterprises to “talk” to each other.
- The communication layer that uses protocols and standards for communication between applications and people from the different enterprises. Application protocols, such as SMTP, HTTP, FTP and SOAP, ensure not only that people from different enterprises can communicate with each other, but that applications from different enterprises can communicate with each other as well.
- The collaboration/coordination layer that uses protocols and standards for collaboration between people and coordination of business processes. The later represents the connection of the processes of the participating companies and is the means for an effective business co-operation.
    - For the collaboration between people, IP-based solutions and standards are partly well established (e.g., SMTP for e-mail, NNTP for news, WebDAV for document sharing, H.323 and T.120 series for real-time voice and data conferencing) and are partly starting to appear (SIP for Internet Telephony, SIMPLE for Presence and Instant Messaging).
    - On the other hand, solutions and standards in the area of co-ordination of business processes are starting to appear (e.g., ebXML Business Collaboration Protocol, Web Services Flow Language [7]), but many issues are still unresolved.
- Finally, the top layer, called the cooperation layer, which deals with searching and finding new partners, negotiating and closing contracts and working together to set the boundaries of their co-operation. Infrastructural support at this level is still very much at the research stage. Research initiatives addressing directory services, such as UDDI and ebXML Registry, electronic contracting and virtual and networked organisations can be classified at this level.

Therefore, achieving full interoperability among two business partners means that they are able to collaborate at all levels of their enterprise architecture. In other words, interoperability does not only address the ability of software components to collaborate regardless of different languages, data formats, interfaces, execution platforms, communication protocols or message formats. A systematic approach to interoperability should also take into account interoperation issues at more abstract levels, such as business process interoperability. Business process interoperability is characterised as the ability of business activities of one party to interact with those of another party, whether or not these business activities belong to different units of the same business or to different businesses.

Some interesting research issues [8], [11], [17] in the area of interoperability among different business partners are presented below:

- Process Ownership: In many interorganizational settings (such as e-government, health care processes, educational programs, service industry) business processes are only sparsely structured and formalized, rather loosely coupled, and/or based on ad hoc cooperation and often there is no explicit or implicit agreement of process ownership. More research is needed to investigate in more detail the dilemmas of distributed versus centralized ownership and bring out different models of interorganizational process ownership to support handling the related issues in an integrated way.

- Reference Model: One of the basic obstacles in interoperability is the lack of a comprehensive model of interorganizational business processes as a basis for contracting and standardization.
- Autonomy and Privacy: an important issue for the enterprises that operate in collaborative environments is the assurance of their local processing privacy and autonomy.
- Heterogeneity-Dynamicity: There is low interoperability among companies due to their heterogeneity in hardware, software, and modelling. Besides, most research in this area assumes that the enterprises that are going to join in a network are homogeneous which is not the case especially when someone refers to Virtual enterprises that have a dynamic character.
- Dynamic processes: Process orchestration is starting to catch the attention of the industry, and as a consequence many competing proposals for process specification languages have been introduced, e.g., ebXML BPSS [18], WSCL [19], WSFL [23], XLANG [24], BPEL4WS [19], BPML [25]. However, neither of the current proposals provides a complete solution, nor do they address the future demand for dynamic processes. Thus, there is a great need for architectures readjusted to dynamic process enactment and execution infrastructures.
- Trust: this issue is vital in dynamic environments where a company needs to continuously establish collaboration with different enterprises and expose its processes and know-how.
- Process Modeling: during collaboration among companies business processes have a distributed nature. Therefore there is a challenge in enforcing transaction semantics over the entire process. In general, extending transaction models to support business processes has attracted considerable amount of research attention [11].
- Exception Handling: it is very important within an inter-enterprise collaboration framework to find ways in order to express and support exceptions that may occur (exception handling) during the execution of a collaborative business process.
- Methods and Tools: these are needed for building a library of reusable and composable common business documents and processes.
- Process Monitoring: a mechanism is needed that will aid not only at the process monitoring procedure in the distributed environment of the collaboration networks but also for the analysing the results.
- Searching/matching mechanisms: efficient registration and searching/matching mechanisms are needed in order for the enterprises to form effectively a collaboration network.
- Human Interaction in Business Processes: It is important to examine the requirements of many inter-enterprise business processes that contain a significant amount of human interactions and examine how traditional collaboration tools that are built around intra-enterprise collaborations (i.e., within a firewall) can be extended to an inter-enterprise environment, and how they can interoperate with inter-enterprise collaborative process management.

During the proposed Ph.D. research, investigation will be focussed on a subset of the issues which are more closely related to the Ph.D. topic as it will formed and specified in the near future.

4 Workflow Management Systems

Business processes can be thought of as a combination of real world models (usually called process models) and technology models (e.g. workflow models) [13]:

- The Process Model describes the steps that occur in the real world (e.g., the travel agent that fulfils a reservation, the ticket availability and payment of the ticket).
- The Workflow Model is the most popular technology for realizing process models. More specifically, a workflow model describes the technology interactions that support, interact with, or implement a real-world Process Model (e.g., reservation system X sends a request to system Y that checks the ticket availability).

Workflow models need Workflow Management Systems for their implementation, i.e. for the implementation of the business processes that have been enacted by business people.

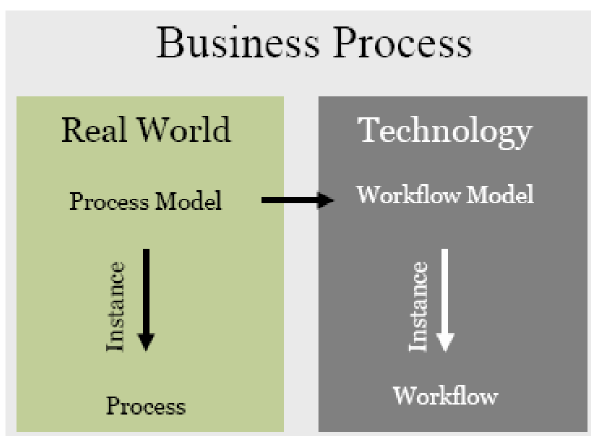


Fig. 2. Business Process = Process Model + Workflow Model [13]

In order to satisfy the needs that exist in a collaborative network, for a number of years the research on workflow management has focused also on interorganizational issues such as [8], [16]:

- Coupling of workflows.
- Contracting of workflows.
- Expanding Petri net application to workflow technology.

- Support through XML based schema definition.
- Web services for workflow.
- Business Process Execution Language for Web Services.
- Use of other standards (as sought by the Workflow Management Coalition).
- Autonomy of local workflow processing.
- Difference in levels of local workflow automation (e.g. degree of implementation and IT support).
- Variation in workflow control policy.
- Confidentiality, which actually prevents complete view of workflow.
- Low interoperability due to heterogeneity in hardware, software, and modelling in multiple organizations.
- Lack of cross-company access to workflow resources.
- Overload of WF servers.

Some of the above issues will be investigated in depth during the proposed Ph.D. research.

5 Virtual Enterprises

Nowadays, enterprises are adapting their traditional business models in favour of collaborative ones that allow them to enter into and exit from markets much faster and at much lower cost. These collaborative business practices, facilitated by technology, are giving rise to new organisational forms that are called Virtual Enterprises (VE). The interoperability of enterprise applications is a central issue in VEs. Although willing to co-operate and inter-operate with others to fulfil the common goals of the VE, every enterprise has its own conditions and requirements. For instance, it may wish to maintain its rights to local choices and solutions (e.g. proprietary enterprise applications), or it may wish to protect or restrict access to its proprietary information.

There are different Virtual enterprises categories [14]:

- Static Virtual Enterprises where partners form a static network with pre-defined process interfaces which are tightly coupled and well integrated. This category may be organized either by a centralized way, where one member of the network coordinates the relationship among the partners, or a decentralized way where business processes are deployed in a jointly, coordinated way.
- Dynamic Virtual Enterprises are also organized either in a centralized or decentralized way and characterized by inconstant business relationships which are mediated by a virtual market place and are evolving and dissolving based on specific criteria.

Dynamic Virtual Enterprises are created by following the steps below [14]:

- A company that desires to join a collaborative network declares its requirements to a marketplace.
- The marketplace provides matchmaking service in order to find candidate companies that are able to offer the services required.

- Last but not least a negotiation process takes place between enterprises which may have a contract as an outcome.

Some of the factors that lead individual companies to form virtual enterprises are: transaction costs, specificity of the product, flexibility, timeliness and costs [14]. Although research communities are focused in forming virtual enterprises many issues are still have not been answered especially when someone refers to dynamic virtual enterprises:

- Finding of required resources in the open market. This relates to issues: trust integration (cultures and systems).
- A framework where the timeframe of the collaboration is only for one project.
- Approaches where the collaboration rely on a leader or dominating broker which caters for resource selection.
- Forms of Virtual Enterprises where all partners are equal also attract the attention.
- The way that contracts are established among the companies that form a Virtual Enterprise.
- Data security.
- A common business process description language is required that should virtually be able to describe any task or process regardless of the industry.
- Mechanisms of change that are necessary within any partner network to allow for rapid configuration are also not yet researched [15].

All the above issues are very important for the effective function of virtual enterprises. During my research, I will focus on the research issues which are closely related to my Ph.D. topic, as it will be specified in more detail in the upcoming months.

6 Proposed Research Plan

During my research I am planning to work on inter-organizational collaboration. In general I am interested in dynamic environments which have specific requirements in terms of interoperability when it comes to forming collaboration networks. More precisely my main focus will be on two of the layers that appear in Fig. 1:

- On the business process layer (co-ordination of business processes) which is concerned with the joint business process among an enterprise network. This means that before the enterprises engage in a transaction they need to agree on the procedures of their joint business process. I am going to do research on the ways that the joint business process can be modelled while preserving the autonomy of the enterprises' private processes. Furthermore I will try to express and support exceptions that may occur (exception handling) during the execution of the joint business process.
- On the upper level of the infrastructure (co-operation between enterprises). Under this framework I will concentrate on how the interactions among the enterprises can be clearly defined, so there is no ambiguity as to what a message may mean, what actions are allowed and what responses are expected using semantics. Interoperability at this higher level is a challenging issue because it requires

an understanding of the semantics of partner business processes [6]. Moreover I will investigate the way contracts can be realised among enterprises in order to enable real-time transactions.

Last but not least, I am planning to investigate the use of web services as the underline technology for implementing my proposed framework.

In the immediate future I am planning to focus on the state-of-the-art of Virtual Enterprises, Workflow Systems and Semantics in order to have an overview of all the research issues in these areas, while I will continue looking for standards that enable the interoperability on the architectural layers of my interest.

Acknowledgement

The author would like to thank her Ph.D. supervisor, Prof. Aphrodite Tsalgatidou, for her support in the development of this research plan.

References

1. McGoveran, D.: An introduction to BPM & BPMS. Business Integration Journal, April (2004)
2. W.M.P. van der Aalst, A.H.M. ter Hofstede, Weske M.: Business Process Management: A Survey. (2003). http://citeseer.nj.nec.com/579893.html
3. European Commission: IST 2003 the Opportunities ahead. (2003)
4. Berre A.J., Hahn A., Akehurst D., Bezivin J., Tsalgatidou A., Vermaut F., Kutvonen L., Linington P. F.: State-of-the art for Interoperability architecture approaches. November 19, (2004)
5. Steen M. and Hofte t.H.: Integrating Collaborative Support and Business Transaction Environments Towards a Research Roadmap for Europe. Telematica Instituut, (2002)
6. Medjahed B., Benatallah B., Bouguettaya A., Ngu A., Elmagarmid A.: Business-to-business interactions: issues and enabling technologies. VLDB Journal, 3 April (2003)
7. Selim A., Pallavi M., Krishnamurthy S.: E-business Process Modeling: The next Big step. IEEE May (2002)
8. Larsen M. H., Klischewski, R.: Process Ownership Challenges in IT-Enabled Transformation of Interorganizational Business Processes. IEEE (2004)
9. McDonald C.: Orchestration, Choreography, and Collaboration.http://www.sun.com
10. European Public Administration Network eGovernment Working Group: Key Principles of an Interoperability Architecture. Presidency of the Europeaen Union, Ireland (2004)
11. Dayal U., Hsu M., Ladin R.: Business Process Coordination: State of the Art, Trends, and Open Issues. 27th VLDB Conference, Roma (2001)
12. Ghalimi I., McGoveran D.: Standards and BPM. Business Integration Journal, April (2004)
13. IONA Technologies PLC: The Seven Principles of Web Services Business Process Management. IONA Technologies. December (2002)
14. Bischofberger, T.: IT-Infrastructure for Dynamic Virtual Enterprises. Master Thesis. January (2004)
15. Katzy B., Löh H.: Virtual Enterprise Research State of the Art and Ways Forward. May-November (2002)

16. Reichert M., Bauer T., Dadam P.: Enterprise-Wide and Cross-Enterprise Workflow Management: Challenges and Research Issues for Adaptive Workflows. Proc. Workshop Informatik (1999)
17. Samir T., Issam C.: A bottom-up Approach to Inter-Enterprise Business Processes. Proc. 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises(WET ICE ’04)
18. ebXML: ebXML Business Process Specification Schema. http://www.ebxml.org
19. WSCL: Web Service Composition Languages. http://tmitwww.tm.tue.nl/research/patterns/
20. Tibco: Tibco Staffware Suite. http://www.tibco.com
21. WebMethods: Business Process Management. http://www.webmethods.com
22. Vitria:BusinessWare. http://www.vitria.com
23. IBM: WSFL. www-3.ibm.com
24. XLANG. http://www.gotdotnet.com
25. BPML. http://www.ebpml.org

Contribution to a methodology to develop interoperability of enterprise applications

Nicolas Daclin[1]

[1] Université Bordeaux 1, LAPS, 351 cours de la Libération
33405 Talence, France
nicolas.daclin@laps.u-bordeaux1.fr

Abstract. This paper aims at presenting an overview on a methodology to develop interoperability of enterprise applications. After a presentation of the definition of interoperability used as starting point and the main objective of this research, approaches to develop interoperability, particularly the federated approach, in the frame of networked enterprise will be given. A draft of interoperability general approach as well as axis of methodological frame which are currently developed will be proposed. Conclusion will be given at the end of the paper.

1 Context and objective to develop interoperability

According to the definition from IDEAS [1] project, interoperability can be defined as the ability of interaction between enterprise software applications. The interoperability is considered achieved if the interaction can, at least, take place at the three levels: *data*, *application* and *business process* with the semantics defined in a business context. The business level is decomposed in three sub levels: *business model*, *decisional model* and *business process.* This definition is represented by the IDEAS interoperability framework as shown figure 1.

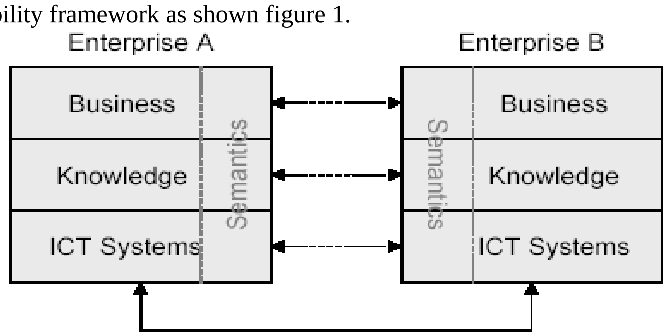


Fig. 1. Simplified IDEAS interoperability framework

In this context, the main objective of the research is to elaborate a methodology allowing establishing interoperability between enterprise applications in the frame of networked enterprise (e.g. virtual enterprises) [2]. This research focuses on interoperability at the level of enterprise modelling (process, decision-making…).

2 Approach and expected research result

According to ISO 14258 [3], it exists three approaches to develop interoperability: integrated, unified and federated. The integrated approach which requires a standard format for all the constituents systems is considered not adapted to networked enterprises environment. The unified approach can only be used if there exists a predefined meta-model for mapping between diverse models/applications to establish semantic equivalence. The federated approach seems most promising to networked enterprises where models and applications need to dynamically adapt and accommodate, in particularly in the virtual enterprises environment. Indeed using federated approach to establish interoperability allows the enterprises in question to keep their own identity, methods of work, tools and applications; and possibly with reduced time, limited changes and costs to establish interoperability.
This research work aims at developing a methodology that provides the guide on how to implementing an interoperability solution in the networked enterprise through a federated approach. More precisely the methodology allows establishing interoperability by: (1) following a structured approach in a step-by-step manner; (2) dynamically composing available interoperability solution elements according to identified requirements; (3) identifying and involving various actors and stakeholders of the enterprises concerned.

3 Interoperability general approach and methodological framework

General approach to achieve interoperability is represented as shown in figure 2:

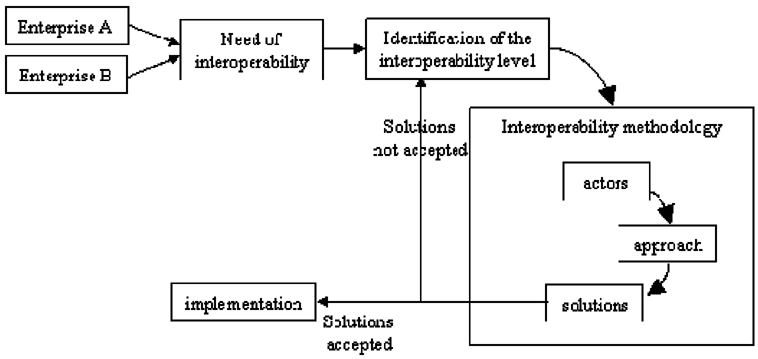


Fig. 2. Interoperability general approach and Methodological frame

A methodological framework will be defined to structure various concerns:

- **Actors axis**: here it is question to determine the actors and stakeholders of the interoperability analysis and design, focusing on the three sub levels. As an example, in the decisional interoperability context it is necessary to identify all decision-markers contributing to the decision making.

- **Approach axis:** the goal is to start from the existing system in order to determine the need of interoperability, the different obstacles to establish interoperability (analysis), and to compose the solution to establish interoperability

- **Solutions axis:** this is concerned with modelling tools, concepts, principles and partial solutions to use for each sub level (*business model*, *decisional model* and *business process*). Actors/stakeholders at a given phase of the approach can choose necessary solution elements to support the modelling, analysis and design.

4 Conclusion

This research develops a methodology to help establishing interoperability at the business level in the context of virtual enterprise. The federated approach is chosen. A methodological framework will be defined to structure various concepts, approaches and solution elements. Currently we focus our research at the decisional model level to study how to improve decisional interoperability.

References

1. IDEAS (2003), IDEAS project Deliverables (WP1-WP7), Public Reports, www.ideas-roadmap.net, 2003.

2. O. Perrin, C. Godart, A model to support collaborative work in virtual enterprises, Data & Knowledge Engineering 50, 2004, pp 63-86.
3. ISO (1999), ISO 14258 - Industrial Automation Systems - Concepts and Rules for Enterprise Models, ISO TC184/SC5/WG1, April 14, 1999.

Adventures in Modeling Spaces: Close Encounters of the Semantic Web and MDA Kinds

Dragan Djuric

University of Belgrade, FON, Jove Ilica 154, 11000 Belgrade, Serbia and Montenegro
dragandj@gmail.com

Abstract. This paper presents the original idea of Modeling Spaces and its impact on understanding various modeling domains and their interconnections. The focus is given on The Semantic Web and Model-Driven Architecture technical spaces and modeling spaces they are built around – RDF(S) and MOF modeling spaces. It also clarifies their mutual relations and helps to understand what is necessary to do to bridge them and acquire their interoperability. This work is a part of the work the author is currently doing for his PhD thesis "Models of The Semantic Web".

1 Introduction

Recent software engineering efforts rely on many concepts like models, metamodels, model transformations, etc. Although most of them try to exemplify most important benefits, they very often do not consider how software practitioners understand modeling. In fact, when talk about models, software engineers often think of a specific kind of models – UML models. However, there are many open questions such as whether we should assume the code we write as a model or not; what are models and metamodels and why do we need them; what means transforming a model into a programming language, etc.

In the past few years, two major approaches to modeling have emerged object-oriented (OO) Model-Driven Architecture (MDA) and ontology-based Semantic Web (SW). In this paper, we show how these competitive approaches can be generalized to the point where their mutual characteristics help us to explain how they can interoperate. Moreover, we present the original idea of Modeling Spaces (MS) [7], a formal framework for the understanding of different modeling approaches, which we then use to explain MDA and SW in a similar way and to identify how their connection can be acquired.

After the introduction, we present a short overview of SW and MDA. Then, we dedicate the third section to the definition and explanation of MSs, as they are new, original, concept. In the fourth section, we give the details of how MS can be used to explain the connection between RDF(S) (ontologies) and MOF (OO). The fifth chapter brings a more precise definition of technical spaces (TS) [11] with the help of MS and the explanation of the connection between SW and MDA TSs. Then, we bring the conclusions and say something about the future work.

2 A short overview of The Semantic Web and Model-Driven Architecture

The step beyond the World Wide Web is the Semantic Web [3], which enables machine-understandable data to be shared across the Net. The Semantic Web is powered by metadata, described by ontologies that give machine-understandable meaning to its data. Ontology is one of the most important concepts in knowledge representation. It can be generally defined as shared formal conceptualization of particular domain [8]. The World Wide Web and XML provide the ontologies with interoperability, and these interoperable ontologies, in return, facilitate Web that can "know" something.

Semantic Web architecture is a functional, non-fixed architecture [2]. Barnes-Lee defined three distinct levels that incrementally introduce expressive primitives: metadata layer, schema layer and logical layer [3]. Languages that support this architecture and the place of OWL are shown in Figure 1.

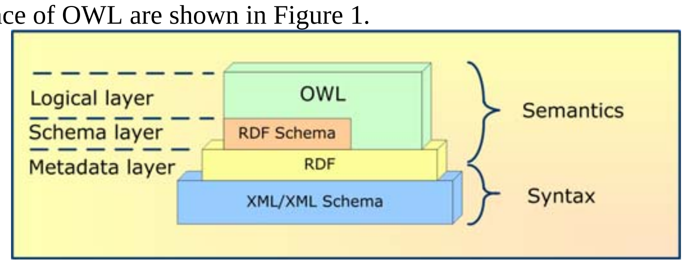


Figure 1. OWL in the Semantic Web architecture

Common data interoperability in present applications is best achieved by using XML. As shown in the Figure 1, XML supports syntax, while semantics is provided by RDF, RDF Schema and mainly by OWL [9]. In order to provide capabilities for unconstrained representation of the Web knowledge and, in the same time, to support calculations and reasoning in finite time with tools that can be built on existing or soon available technologies, OWL introduces three increasingly expressive sublanguages for various purposes: OWL Full (maximal expressiveness), OWL DL (guaranties computational completeness) and OWL Lite (for starters).

Model Driven Architecture (MDA) defines three viewpoints (levels of abstraction) from which a certain system can be analyzed. Starting from a specific viewpoint, we can define the system representation (viewpoint model). The representations/models/viewpoints are Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) [12]. MDA is based on a four-layer metamodeling architecture and several complementing OMG standards, Figure 2. These standards are Meta-Object Facility (MOF) [MOF 2003], Unified Modeling Language (UML) [14] and XML Metadata Interchange (XMI) [12], and the layers are: meta-metamodel layer (M3), metamodel layer (M2), model layer (M1), and the real world layer (M0).

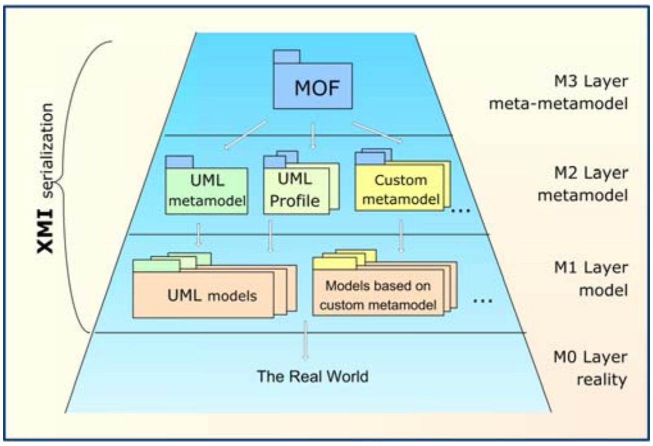


Figure 2 - MDA four-layer MOF-based metadata architecture

The topmost layer in this architecture (meta-metamodel, MOF) defines an abstract language and framework for specifying, constructing and managing technology-neutral metamodels. It is the foundation for defining any modeling language, such as UML or even MOF itself. All metamodels (both standard and custom) defined by MOF are positioned at the M2 layer. The models of the real world, represented by concepts defined in the corresponding metamodel at the M2 layer (e.g. UML meta-model) are at the M1 layer. Finally, at the M0 layer are things from the real world. The purpose of the four layers with common meta-metamodel is to support multiple metamodels and models and their scaling – to enable their extensibility, integration and generic model and metamodel management.

3 Modeling Spaces essentials

The most fundamental definition of model is that it is the simplified abstraction of reality [15]. This definition applies not only to models in technology, but in art or everyday life. Having this definition in mind, we can draw two important conclusions. First, something can be taken as a model if it is an abstraction of things from the real world, but it is simultaneously a thing from the real world. Whether we take it as a model or as a real/world thing depends on the context, i.e. on the point of view. Second, models can be defined using metamodelling concepts formally or implicitly. Since implicit metamodels cannot be precisely defined using formalisms, as in the case of art, in the rest of this discussion we analyze only formal models. Nevertheless, much of the conclusions can also be applied to implicit metamodels.

Figure 3 shows a general modeling architecture that was inspired by MDA and is in fact its generalization. In such a modeling architecture, the M0 layer is the real world as in [4] and [1]. It includes all possible things that we try to represent using the models residing at the M1 layer. That representation is more or less abstract and simplified, depending on how rich our models are. Models are defined using concepts defined in metamodels, so each metamodel determines how expressive its models can be. M2 is the layer where the metamodels are located. The metamodels are also defined using some concepts. A set of concepts used to define metamodels resides at the separate M3 layer at the top of this architecture and is called meta-metamodel. Meta-metamodel is nothing more than a metamodel that is conventionally elected to be used for other metamodels' definition; it also defines itself. The architecture is generalized to comprise not only models and metamodels based on an object-oriented meta-metamodel like MOF is, but also other systems, for instance: ontologies, Semantic Web technologies or non-technical representations.

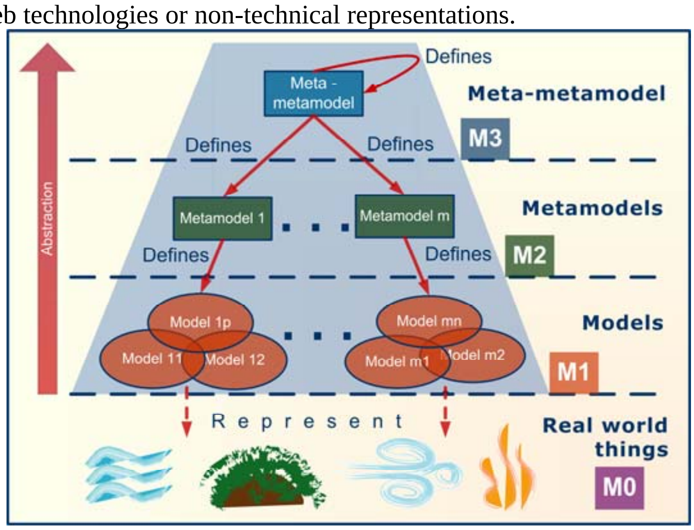


Figure 3 – General four-layer modeling architecture

This is a convenient place to introduce the concept of modeling spaces. A modeling space (MS) is a modeling architecture defined by a particular meta-metamodel. Metamodels defined by the meta-metamodel and models defined by those metamodels represent the real world from one point of view, i.e. from the point of view of that MS. As the meta-metamodel defines the core concepts used in defining all other metamodeling concepts, it is defined by itself. If it was defined by some other concepts, it would not be a meta-metamodel, it would be an ordinary metamodel in some other MS.

Figure 4 shows a few examples of well-known MSs. The most straightforward example from this picture is the MOF MS. It is defined by the MOF meta-metamodel, which in turn is self-defined. It defines various metamodels, for instance Unified Modeling Language [14] or Ontology Definition Metamodel [6], that are used to describe models that represent things from the real world. The same reality is described in the context of other MSs, like RDF(S) or EBNF spaces. Many software engineers would associate the terms like model and modeling exclusively with UML aristocracy, taking EBNF-based models (Java, C#, C++ code) as more technical, flattened artifacts and ignoble citizens. However, Java (or C++, or some other) code is a model, since it represents some simplified abstraction of reality. The same is with

XML code, databases, books, etc – they are all models, but modeled in terms of different MSs, defined by different meta-metamodels.

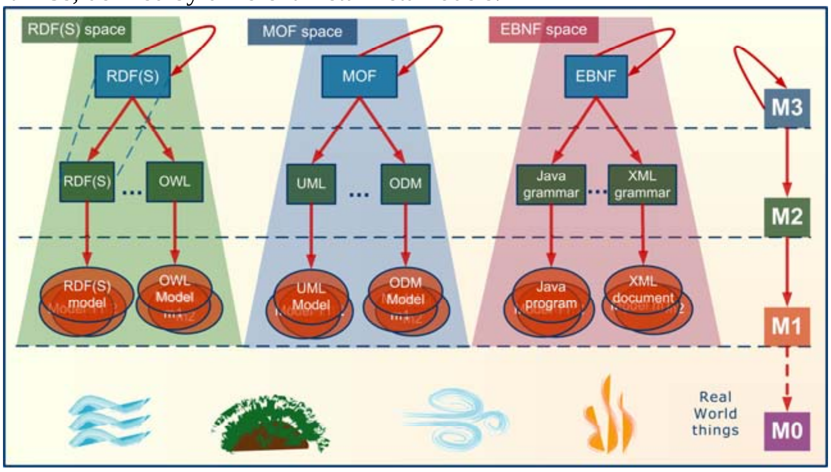


Figure 4 - RDF(S), MOF and EBNF modeling spaces

If we model the real world in a certain MS, we will use some models. If we model the same reality in another MS, we will describe it with different kinds of models, highlighting other characteristics when abstracting from reality. The models from the first MS will be a part of reality that we can model using the models from the second MS. Figure 5 clarifies this duality by an example of the same thing being simultaneously a model and a real-world thing. Along the vertical axis, the world is modeled in the MOF MS. Along the horizontal axis is the EBNF space hierarchy, which is a real-world thing in the MOF space. An interesting observation here is that any MS, like the EBNF space or even the MOF space itself, is a part of the real world from the MOF-space point of view. In general, the way we model some business system or another "real" domain is pretty much the same as the way we model meta-metamodels, metamodels or models from another MS. Of course, these models involve a certain level of abstraction, so there is a possibility of losing some information.

For many software engineers, this duality is complicated to understand at first. The fact that M1-M3 layers are fiction and above the M0 layer does not mean that meta-metamodels, metamodels and models are things outside of reality. Everything is in the real world; we just use a convention to put some things in layers, depending on the context.

MSs can be defined in more or less abstract manner. Some MSs are focused on conceptual (abstract or semantic) things, like models, ontologies, mathematical logics, etc. They are not interested in techniques for representation or sharing their abstractions. We call them conceptual MSs. However, we must have some techniques to materialize (or serialize) those MSs. We can do this using concrete MSs, which are equipped with syntax. Examples of those materializations are some syntax or databases.

Being able to represent bare syntax, concrete MSs need a means to express the semantics, i.e. the meaning of the data they carry. Conceptual MSs, on the other hand, are able to represent semantics, but need a means to represent their information physically. It is obvious that they should complement each other's representation abilities to create models that have both semantics and syntax. One of the most interesting examples of this symbiosis of various MSs can be found in OMG Model Driven Architecture. An example of a conceptual space is MOF MS, while an example of concrete MS is EBNF MS.

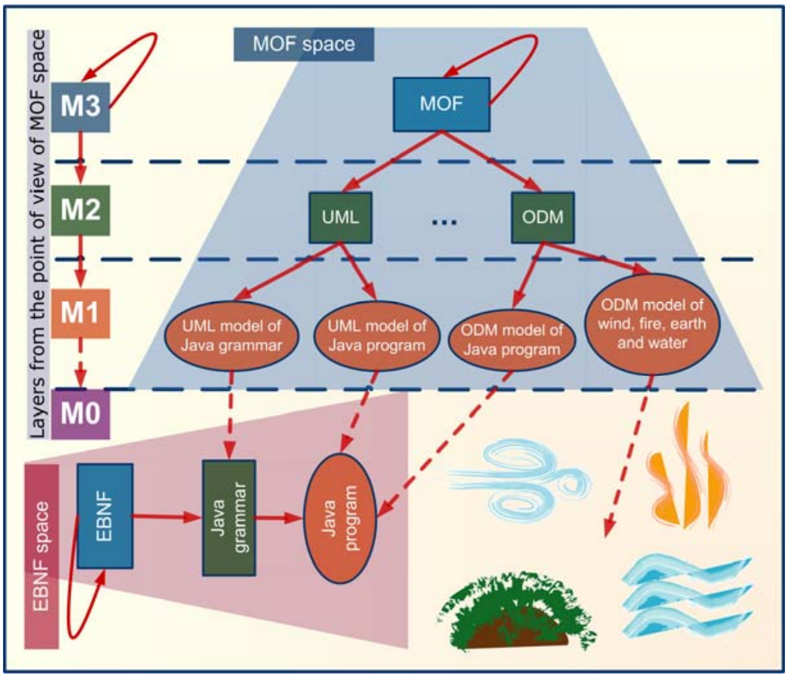


Figure 5 – MOF MS sees EBNF MS as a set of things from the real world

There are two types of usage scenarios for different MSs:

Parallel spaces – one MS models the same set of real-world things as another MS, but in another way. In this case, the relation between these MSs is oriented towards pure transformation, bridging from one space to another. Examples of such parallel MSs are MOF and RDF(S) MSs.

Orthogonal spaces – some MS models concepts from another MS, taking them as real-world things, i.e. one MS is represented in another MS. This relation is often used in round-trip engineering to facilitate different stages of modeling some system. For example, in order to make a Java program we could first use Java UML profile to create classes and method bodies, then transform this UML model into Java code, and complete the Java program using a Java IDE. Orthogonal MSs are also used when a conceptual MS is implemented using a certain concrete MS – for example, when one develops a MOF-based repository to run in a Java virtual machine.

4 The Touch of RDF(S) and MOF modeling spaces

Usage scenarios for parallel spaces most often pertain to conceptual MSs that model the same reality using different concepts. Each of these MSs is implemented in some other, more concrete MS, as a represented reality. In order to exchange models between conceptual MSs, it is necessary to provide transformations from one space to another. These transformations are also models [5], and should be developed in a MS that can represent both the source and the target MSs. Moreover, the transformation also has to be represented in some concrete MS orthogonal to the source and the target MSs, that leads the conceptual model of transformation to its implementation.

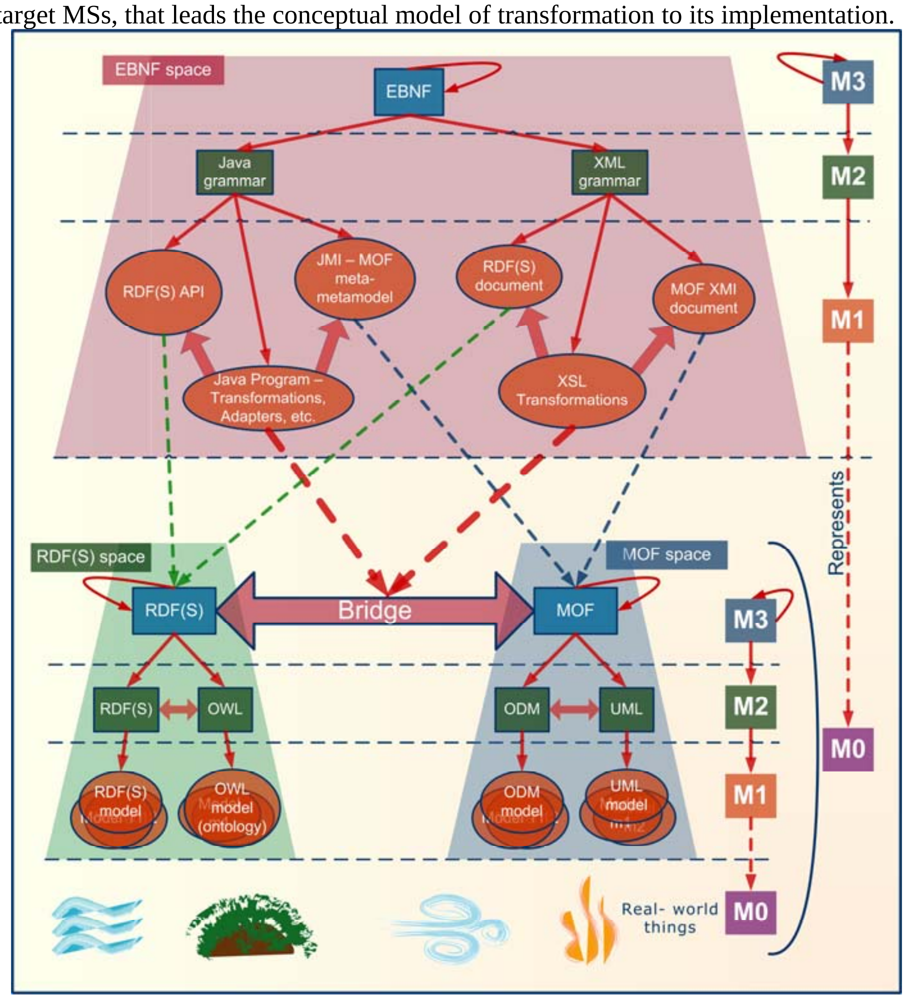


Figure 6 – Transformations between RDF(S) MS and MOF MS

Figure 6 shows two parallel conceptual MSs, RDF(S) MS and MOF MS, and the space that represents them orthogonally, EBNF MS. MOF and RDF(S) model the real world in parallel, using some modeling languages (UML, ODM, OWL or other) that are defined using different meta-meta concepts. At the conceptual level, we could establish a transformation from one language to another, e.g. UML to ODM and vice versa, in the same MS. An example of a transformation modeling language for such purposes in MOF is Query-View-Transformation (QVT) [MOF QVT 2002]. RDF and RDF Schema, and three different dialects of OWL – OWL Full, OWL DL and OWL Lite are examples of languages at M2 layer of RDF(S) MS. Efforts to develop query and transformation language in RDF(S) MS are underway: Triple, RQL etc. [10].

We can also establish a transformation between MSs, a bridge that transforms RDF(S) concepts to MOF concepts at M3 layer. Using that bridge, we can transform any metamodel (language) defined in RDF(S) into its corresponding metamodel in defined in MOF (both are at M2 layer). Of course, we could expect an information loss depending on how meta-meta concepts (M3) are similar. RDF(S) concepts, rdfs:Class and rdf:Property are similar, but not the same as MOF Class, Association and Attribute.

Both MOF and RDF(S) spaces can be represented in other, more concrete MSs. They can be implemented using repositories, serialized into XML etc., which involves many MSs. For the sake of simplicity, we have skipped a few steps and have shown them as Java program codes and XML documents in the EBNF space. Models from the MOF space are modeled in Java code according to JMI standard, and in XML according to the MOF XMI. For languages from RDF(S), XML is the natural way of representation. They can also be modeled using Java APIs (Jena etc.).

As the RDF(S)-MOF bridge is also a model; it can be also represented in a concrete MS representing meta-metamodels that should be bridged. Examples include an XSLT that transforms a MOF XMI document into RDF(S) XML document and vice versa, a set of Java classes that adapt JMI interfaces to RDF(S) API interfaces, or a Java program that does a batch transformation from a JMI-based code to an RDF(S) API-based one.

As Figure 6 shows explicitly, a single bridge models a transformation between two MSs at layer M3, between RDF(S) and MOF meta-metamodels. Transformations between metamodels situated in a single MS at M2 layer are internal to that MS. However, they can be implemented through some concrete MSs (e.g. EBNF for XSLT).

5 The Touch of the Semantic Web and MDA technical spaces

MS is a concept inspired by the concept of TS, which is defined as a working context with a set of additional concepts, body of knowledge, tools, required skills, and possibilities [11]. Fortunately, we can use MS to enhance this fuzzy definition of TS.

A technical space (TS) is a working context that includes various related MSs. Most often the TS is built around some MS, whereas the role of other MSs is supportive (e.g., implementation), or implicit (literature, know-how). For example, the MOF MS is at the center of the MDA TS. However, the MDA TS also partially includes other MSs: XML and EBNF in the area of XMI representation, EBNF in the area of repository implementation (JMI), an implicit MS that includes literature, etc. Transformations, for example to plain Java, C++ or VB code, are also models belonging to one or several MSs that are partially included in the MDA TS.

Figure 7 shows overlapping between the MDA TS and the Semantic Web TS in some MSs (most MSs belonging to these TSs are omitted for the sake of simplicity). The MDA TS is built around the MOF MS, which resides completely in the MDA TS. The MDA TS also includes OWL ontologies that model MOF-based concepts or contain some knowledge about MDA, which are parts of the RDF(S) MS. On the other hand, SW TS includes RDF(S) MS. Additionally, it includes parts of the MOF MS related to ODM metamodel and Ontology UML Profile, and two-way transformations from these MOF-based models to OWL. These transformations are also a part of the MDA TS. Recall that those transformations are also modeled, so they belong to some MSs as well. Some researches are trying to identify a way to enable transformations between different MSs at the M3 layer using just one two-way transformation for all three layers [4].

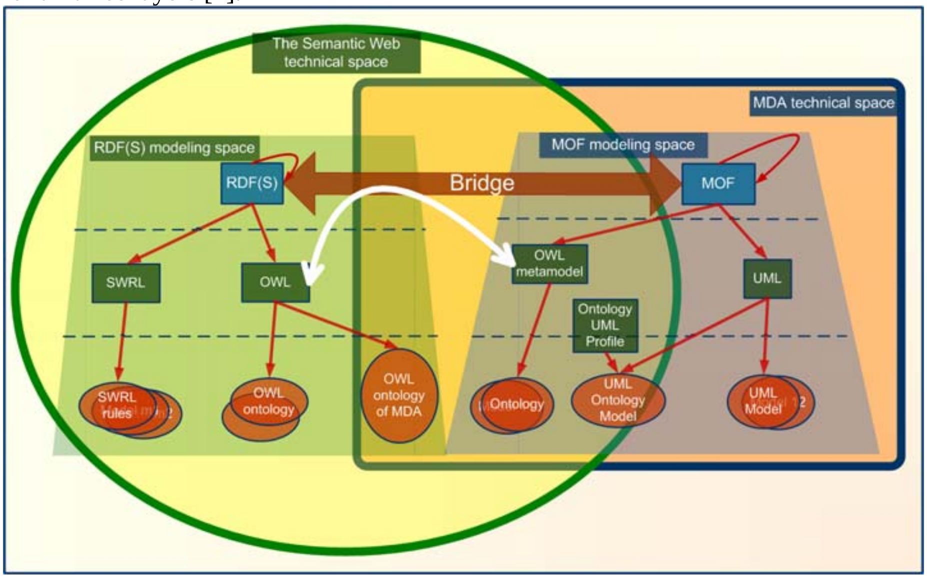


Figure 7 – The Semantic Web and Model Driven Architecture technical spaces

It follows from the above discussion that a TS includes one or more MSs and that each MS is a part of one or more TSs, whereas TS is a means for grouping MSs that have something in common or simply need to interact. The bridge connecting two MSs is also a means for connecting surrounding TSs.

6 Conclusions

We have presented here a concept of modeling spaces and used it in the definition of technical spaces and the explanation of two significant MSs – RDF(S) (from SW TS) and MOF (from MDA TS). We showed how models from RDF(S) MS are viewed in MOF MS and vice versa. The SW and MDA interoperability issues are also shown with the explanation how it can be acquired from the point of view of MSs using bridges between meta-meta models.

An interesting thing is that MSs are suitable to explain existing modeling architectures, even non-technical, and their implementations (MDA, Ontologies, EMF, EBNF, XML, etc.). Being in accordance with existing efforts, it is a suitable means for their clarification, explanation, and their conceptual connection.

References

[1] Atkinson, C., Kühne, T., “Model-Driven Development: A Metamodeling Foundation” (Spec. issue on Model-Driven Development), IEEE Software, Vol. 20, No. 5, Sep/Oct, 2003, pp 36-41.

[2] Berners-Lee, T., “Semantic Web Road Map”, W3C Design Issues, http://www.w3.org/DesignIssues/Semantic.html, 1998.

[3] Berners-Lee, T., Weaving the Web, Orion Business Books, London, 1999.

[4] Bezivin, J., et al “A M3-Neutral infrastructure for bridging model engineering and ontology engineering,” 1st International Conference on Interoperability of Enterprise Software and Applications, Geneva, Switzerland, 2005 (submitted).

[5] Bézivin, J., Dupé, G., Jouault, F., Pitette, G., Rougui, J.E., “First experiments with the ATL model transformation language: Transforming XSLT into XQuery,” In Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of Model Driven Architecture, Anaheim, CA, USA, 2003.

[6] Djuric, D., Gasevic, D., Devedzic, V., “Ontology Modeling and MDA”, in Journal of Object Technology, vol. 4, no. 1, January-February 2005, pp. 109-128. http://www.jot.fm/issues/issue_2005_01/article3

[7] Djuric, D., Gasevic, D., Devedzic, V., “Modeling Spaces”, submitted to IEEE Software, 2005.

[8] Gruber, T. R., “A translation approach to portable ontology specifications”, Knowledge Acquisition, Vol. 5, No. 2, 1993.

[9] Dean, M., Schrieber, G. (eds.), “OWL Web Ontology Language Reference”, W3C Recommendation 10 February 2004 http://www.w3.org/TR/2004/REC-owl-ref-20040210/

[10] Hase, P., et. al., “A comparison of RDF query language proposals”, Proc. Of The 3rd ISWC, Hiroshima, Japan, 7-11 November 2004. http://xobjects.seu.edu.cn/resource/ISWC2004/pdf/32980502.pdf

[11] Kurtev, I., Bézivin, J., Aksit, M. “Technological Spaces: An Initial Appraisal”, CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine, 2002 http://wwwhome.cs.utwente.nl/~kurtev/files/TechnologicalSpaces.pdf

[12] Miller, J., Mukerji, J. (eds.), "MDA Guide Version 1.0.1", OMG Document: www.omg.org/docs/omg/03-06-01.pdf, June 2003.

[13] OMG spec., “Meta Object Facility (MOF) 2.0 Core Specification version 2.0 Final Adopted Specification” http://www.omg.org/cgi-bin/apps/doc?ptc/03-10-04.pdf, October 2003.

[14] OMG spec., “Unified Modeling Language: Superstructure, version 2.0, Final Adopted Specification”, http://www.omg.org/cgi-bin/apps/doc?ptc/03-08-02.zip, August 2003.

[15] Hagget, P., Chorley, R. J. Models, Paradigms and New Geography, In Models in Geography, London, Methuen & Co., 1967.

Synopsis of PhD Dissertation
Modeling of Constraints in Distributed Object Oriented Environment

Chhanda Ray
R.C.C Institute of Information Technology
Canal South Road, Beliaghata, Calcutta-15, INDIA.
e-mail: chhandaray@hotmail.com

Abstract

Integrating data from many heterogeneous sources require interoperability between the data repositories. Object Oriented Data Modeling is gaining popularity because of its elegance in representing real world situations. On the other hand, Integrity constraints traditionally form an essential part of schema definition. In this context, Modeling of constraints in Object Oriented Model has been the focus of attention of many researchers in recent times. We have introduced a constraint model for object-oriented data. We have formalized the constraint model by expressing it using Unified Modeling Language (UML) and ODMG's ODL. To increase interoperability, the constraint model is also represented by using a standard web based interoperable medium like extensible Markup Language (XML). For efficient manipulation of data into a more manageable model that is native to the programming language, an efficient way of storing XML data with constraint information into Java Object Oriented Programming Language is also introduced. Finally, a Wrapper based translator has been developed to accomplish two-way interoperability between XML and Java Source Code towards this end.

Keywords: Interoperability, Object Oriented Databases, ODL, UML, XML.

1.Introduction

Object Oriented data Modeling is gaining popularity because of its elegance in representing real world situations [2], [4]. Methods and tools, such as UML [10], for Object Oriented Modeling are being developed. Object Oriented Databases are also gaining focus over relational ones.

Integrity constraints traditionally form an essential part of schema definition. In addition, they are useful for query optimization, update anomaly prevention and for information preservation in data integration. Integrity constraints are also used to model references in relational databases, through keys and foreign keys. In this context, Modeling of constraints in Object Oriented Data is an active research topic [1], [5], [6], [7], [9], [10], [17]. In [6], an extension of XML DTD has been proposed that specifies both semantic structure as well as integrity constraints for XML data. A Unified Constraint Model (UCM) is proposed in [7] using XML DTD which is both simple and expressive. However, XML Schema Definition language is more powerful than XML DTD for semantic specifications.

Integrating data from many heterogeneous resources require interoperability between the data repositories. Due to increasing popularity of XML as a standard interoperable medium for exchanging data between web applications, there has been a lot of research for transformation of data from various sources including RDBMS repositories to XML and vice versa [3], [15], [16]. An Object Oriented Model for interoperability is introduced in [18] to solve the data and operation inconsistency problem in legacy systems.

In this work, our contribution is as follows. First, we have proposed a constraint model for Object Oriented Data in [11]. The syntax and semantics of such modeling is extended to deal with constraints in single inheritance. The first step of semantic modeling is to identify useful semantic concepts. The useful semantic concepts, then, are captured in formal modeling languages such as UML or ODMG's ODL. Therefore, we have formalized the constraint model by expressing it using UML and ODL in [12]. To increase interoperability, in [13], the constraint model is also represented by using a standard web based medium like extensible Markup Language (XML). For efficient manipulation of data into a more manageable model that is native to the programming language, an efficient way of storing XML data with constraint information into Java Object Oriented Programming Language is introduced in [14]. Finally, a wrapper-based translator has been developed to accomplish two-way interoperability between XML and Java Source Code.

The layout of this paper is as follows. The next section describes the constraint model and Section 3 introduces the formalization of the model using UML and ODL. The derivation of ODMG database design from UML class diagram has been represented in Section 3.1. In Section 4, Representation of XML data with constraints is introduced. An example is also illustrated in Section 4.1. Section 5 describes the interoperability between XML and Java with constraint information and we conclude in Section 6.

2. Modeling Constraints

An integrity constraint is semantic information in an object or a relationship among objects. A constraint specifies a condition and a proposition that must be maintained as true. We attempted to model each constraint as a Boolean method, which returns either true value or false value. If the predicate within a method is satisfied by a model element then the method will return true, otherwise the method returns false. These methods are clearly distinguished from the usual methods of a class by their usage and therefore, the constraint methods are accommodated differently in object oriented data model.

We have considered three different types of constraints present in an object oriented data model - Single attribute constraints, Multiple attributes constraints and Class constraints. Single attribute constraints or constraint attributes are applicable to individual attributes of a class. For example, suppose that there is a class “Employee” with usual attributes and methods. A constraint on the attribute “age” of any “Employee” object may be described as “age of an employee must between 20 & 60”. Multiple attributes constraints or constraint methods involve more than one attribute of a class. For example, there could be a constraint described as “if experience of an employee is less than 5 years then the salary can not be more than $4500”. Class constraints are applicable to all objects of individual classes of an object oriented database system. A typical example of a class constraint in the “Employee” class may be described as “id of an employee must be unique”. To handle single and multi attribute constraints, the element “class” is defined as a 4-tuple <A, CA, M, CM> where A & M represent attributes and methods of a class and CA, CM represent constraint attributes and constraint methods.

To represent class constraints, we introduced a singleton collection class associated with each general user-defined class, where the singleton collection class would always contain a collection of all objects of the user-defined class. All constraints that need to check all objects of the user-defined class for validation become Boolean methods of the singleton class. For example, for the class “Employee”, a collection class “Employee_Collection” may be defined. The class “Employee_Collection” has two attributes, a pointer to the list of all existing “Employee” objects and the total number of “Employee” objects. Whenever a new object of the class “Employee” is created, a reference to that object is added to the list “allElements” and the integer “noOfElements” is incremented. Similarly, whenever, an “Employee” object is deleted then the reference of that object is deleted from the list “allElements” and the integer “noOfElements” is decremented. The collection class also contains all the boolean methods to check the class constraint mentioned for the class “Employee”.

3.Expressing Constraint Model in UML and ODL

The Unified Modeling Language (UML) [10] is the result of an effort in developing a single standardized language for object-oriented modeling. The UML meta model is represented by a Foundation Package that contains all of the constructs provided by UML for modeling software systems. The UML Foundation Package is made up of three sub packages among which the Core Package defines the basic abstract and concrete constructs needed for the development of the object models. As the constraint model [11] enhances the definition of a Class, the UML meta model needs to be extended to accommodate constraints. We introduced four new model elements to the core package of UML meta model is as follows.

a) *ConstrainedAttribute – ConstrainedAttribute* class is introduced as a subclass of *Attribute* class, which is also a subclass of *StructuralFeature* class. *ConstrainedAttribute* class can contains a number of methods. So, this new model element *ConstrainedAttribute* class is connected with the UML meta model class *Method*, which is a subclass of *BehaviouralFeature* class. In UML meta model, *Class* class is a subclass of *Classifier* class which describes behavioral and structural features and thus, each *ConstrainedAttribute* class is associated with *Class* class. The *ConstrainedAttribute* class extends the UML meta model *Attribute* class. By the virtue of inheritance all the attributes of the *Attribute* class, such as *Changeable, frozen, addOnly, initialValue, multiplicity* etc. will be inherited into this class. In addition, it has three new attributes *name, attributeNames, methodCount,* which specifies the name of the constraint, the names of the attributes associated with the constraint and the total number of constraints.

b) *ConstraintedMethod* - The *ConstrainedMethod* class extends the *Method* class of the UML meta model. In addition to all attributes of the *Method* class of UML meta model, it has four extra attributes *name, attributeName, methodCount* and *attributeCount*. *Name* attribute indicates the name of the constraint, *attributeName* is an array of name of the attributes associated with a constraint, *methodCount* represent the total number of constrained methods and *attributeCount* represent the number of attributes associated with a constraint. The *attributeCount* attribute represents the number of elements within the *attributeName* array.

c) *CollectionClass* – The *CollectionClass* class is introduced as a subclass of *Classifier* class of the UML meta model. In addition to all attributes of *Classifier* class, it has three new attributes *className, objectTotal and constraintCount* which represent the name of the user-defined class associated with the *CollectionClass* class, the total number of objects within the user-defined class and the total number of constraints of the user-defined class respectively.

d) *MyClass* – *MyClass* is introduced as a subclass of *Class* class in UML meta model. By the virtue of object orientation, all the attributes of *Class* class will be inherited into *MyClass* class. In addition, it has one new attribute *cclassName,* which represents the name of the *CollectionClass* class associated with the user-defined (*MyClass)* class.

The motivation behind expressing constraint model in UML is formalization. It is expected that such formalization would lead to the development of sound tools for analysis and synthesis.

After conceptual modeling, the model is transformed to a database design (represented by a database schema definition), which can be implemented in an object oriented database (OODB) system. The language used to define the specification of object types for the ODMG/OM is called Object Definition Language (ODL). Extensions to the ODMG/OM to accommodate constraints are discussed in the following.

(i) *Single attribute constraints & multiple attribute constraints* – We regard these constraints definition as a part of class definition and extend the class declaration accordingly.

<class>:: = <class header>{<interface_body>[<cons_dcl>]}
< cons_dcl>::= <cons_spec><attributes_lists>:<action>
<cons_spec>::= constrainedattribute|constrainedmethod
<attribute_lists>::={attr1, attr2,…..attrn}
<action>::= <user_defined_procedure>

(ii) *Class constraints* - To show the connection between a general purpose class and the corresponding collection class we use a *relationship* with in the class definition of the general purpose class. To include the integrity definitions in the ODMG ODL, the BNF for relationship declaration is extended accordingly.

<rel_dcl>:= relationship<collection_type><class_name><relationship_name>
inverse <inverse_class_name>
<collection_type>:= set|bag|list|array|dictionary

To declare the constrained methods in collection class the corresponding BNF is as follows.

<class>:: = <class header>{<interface_body>[<cons_dcl>]}
< cons_dcl>::= <cons_spec><subclass attributes_lists>:<action>
<cons_spec>::= constrainedmethod
<subclassattribute_lists>::={subclassname.attr1,subclassname.attr2,…., subclassname..attrn }
<action>::= <user_defined_procedure>

3.1 ODMG Database Design from UML Class Diagram

The derivation of an ODMG schema begins from a package. Within a package, the following steps are needed to map a UML class diagram to an ODMG-ODL schema:

For each class in UML, create a class definition in ODMG-ODL with the same class name. Then examine the UML class specification:

(i) if the *isAbstract* attribute is false, then add an *extent* definition to the ODMG class. The name of the extent is the plural name of the class name.
(ii) For each attribute in the UML class with cardinality of the form m..1 (m>=0), define the attribute as single value; otherwise, define the attribute as a set.
(iii) For each operation in the UML class, make a correspondent operation definition in the ODL class.
(iv) For each constraint of the UML class:
  - If the name of the constraint is *constrainedattribute*, then first make a correspondent attribute definition in the ODL class and with this definition define an operation with unique name. The return type of the operation is Boolean and the input parameters of the operation is the name of the attribute.
  - If the name of the constraint is *constrainedmethod*, then make a correspondent operation in the ODL class. The return type of the operation is Boolean and the input parameters are the names of the attributes associated with the constraint.
  - If the name of the constraint is class constraint, then create a correspondent relationship definition in the ODMG-ODL class. For example,

    relationship set<Employee>contained_in inverse
    Employee::Employee_collection;

  The related collection class would be define by using the similar class definition as general purpose class.

(v) For each interface in UML, create an interface definition in ODMG-ODL with the same interface name.
(vi) For each generalization between a supertype stype and one or more subtype type1, type2, …………
  - If stype is a class, then add the EXTENDS relationship from type1, type2, …. to stype.
    type1 extends stype
    type2 extends stype
  - If stype is an interface, then add the ISA relationship from type1, type2, ….. to stype.
    type1: stype
    type2: stype
    ……………

The correspondence between the extended UML meta model and a standard Object database model developed by ODMG has been established in [12].

4. Representation of constraints using XML

Retrieval and validation techniques developed for XML documents make it a good candidate for retrieval of Object Oriented Framework. It offers a convenient syntax for representing data from heterogeneous sources. To increase interoperability, we have described a mapping scheme from classes and objects with constraints information to XML Schema and XML data in [13]. A brief description of mapping of classes with constraints to XML Schema is as follows.

(i) For each class,
  - Define class as *complexType* in XML Schema.
  - Define attributes of “class” as *Attributes* of the complexType .
  - Define Data Type of attributes as built-in or user-defined types in XML Schema.

- Introduce an additional attribute cStatus, which indicates whether the complexType represents a general purpose class or a collection class. Set the appropriate value gClass or colClass for representing the above.

(ii) We can handle single attribute constraints in two different ways:

a) Using built-in facilities in XML Schema.
   - Create a *SimpleType* for each constraint.
   - Derive the *SimpleType* from *base type* by *restriction.*
   - Set the range of values for numeric *SimpleType* using the components *minInclusive, minExclusive, maxInclusive, maxExclusive* etc.
   - For nonnumeric *simpleType* set the pattern of values using *Pattern* component.
   - The attribute which has a single constraint must be of the type representing the constraint.

b) Representing constraint methods as attributes
   - Create a *complexType* for each constraint.
   - Derive the *complexType* from *base type* by *restriction.*
   - Introduce the attribute(s) of the class associated with the constraint as *Attribute(s)* of the *complextype*.
   - Enumerate with a single value, where the value is the code of the constraint method.
   - Introduce an additional attribute cType, which indicates whether the constraint represents a single attribute constraint, a multi attribute constraint or a class constraint. Set the appropriate value cAttribute, cMethod or cClass for representing the above.
   - Introduce the name of the class as an additional attribute of the complexType and enumerate with a single value where the value is the class name.

(iii) We can represent each constraint method (multi-attribute constraint) as attribute as described in 4.(ii)b).

(iv) For each class constraint,
- Introduce a constraint method into singleton collection class associated with the user-defined class.
- Each collection class along with attributes and constraint methods would be represented as *ComplexType* in XML Schema using the same convention as described earlier for other user-defined classes.
- Introduce an additional attribute cStatus, which indicates whether the complexType represents a general purpose class or a collection class. Set the appropriate value gClass, or colClass for representing the above.
- Introduce the name of the general purpose class as an additional attribute of the complexType and enumerate with a single value where the value is the class name.
- The attributes and the methods of this class are adequately generic and hence can be handled in an independent manner.

During interoperability, special care must be taken about the constraint methods represented as attributes in XML schema.

4.1 An Example of XML Schema with Constraints

This section describes an example demonstrating the mapping scheme described in the previous section, by translating the “Employee” Class (with Constraints) to XML Schema. For simplicity, we have taken one single attribute constraint and one multi-attribute constraint out of those provided.

(ii) First, we define a simple type for the Single Attribute Constraints “idC” which represents “Employee Id must be greater than zero”.

```
<xs:schema xmlns:xs=“http://www.w3.org/2001/XMLSchema”>
<xs:simpleType name = “idC”> <xs:restriction base = “xs:integer”>
<xs:minExclusive value = “0”/>
</xs:restriction> </xs:simpleType>
```

```
<xs:simpleType name = “status”> <xs:restriction base = “xs:string”>
<xs:enumeration value = “gClass”/>
</xs:restriction> </xs:simpleType>
```

Correspondingly, the XML Schema for the class “Employee” becomes as follows.

```
<xs:complexType name = “Employee”>
<xs:complexContent>
<xs: attribute name = “empId” type = “xs: idC” use = “required” />
<xs: attribute name = “eName” type = “xs:string” use=“required”/>
<xs: attribute name = “eDesig” type = “xs:string” use=“required”/>
<xs: attribute name = “empAge” type = “xs:int”  use = “required”/>
 <xs: attribute name = “salary” type = “xs:float”  use = “required”/>
<xs:attribute name = “experienc” type =”xs:byte” use = “required”/>
<xs:attribute name = “cStatus”        type =”xs:status” use = ”required” />
</xs:complexContent> </xs:complexType>
```

(ii) We introduce a multi-attribute constraint “empC1” which represents “Employee salary can not be more than $2500 if experience of an employee is less than 5 years”. The translation of the constraint “empC1” is shown in the following.

```
<xs:simpleType name = “conType”><xs:restriction base = “xs:string”>
<xs:enumeration value = “cMethod”/>
</xs:restriction> </xs:simpleType>

<xs:simpleType name = “cName”> <xs:restriction base = “xs:string”>
<xs:enumeration value = “Employee”/>
</xs:restriction> </xs:simpleType>

<xs.complextype name = “empC1”>
<xs:complexContent>
<xs:attribute name= “experienc” type = “xs:byte” use = “required”/>
<xs: attribute name = “salary” type = “xs:string” use = “required”/>
<xs:attribute name = “cType” type = “xs:conType” />
<xs:attribute name = “className” type = “xs:cName”/>
<xs:simpleType name = “value”> <xs:restriction base = “xs:string”>
<xs:enumeration value = “body Code for the constraint Boolean method ”/>
</xs:restriction> </xs:simpleType>
</xs:complexContent></xs:complexType>
```

The resulting XML schema for the class “Employee has one more attribute “C1” of the type “empC1” as presented next.

```
<xs: complexType name = “Employee”>
<xs:complexContent>
<xs: attribute name = “empId” type = “xs:idC”  use = “required” />
…………………………
<xs:attribute name = “cStatus” type=”xs:status” use =”required”  />
<xs:attribute name = “C1” type=”xs:empC1”/>
</xs:complexContent> </xs:complexType>
```

The XML application has to identify that the attribute named “C1” is not a general attribute but “C1” denotes an integrity constraint. This can be ensured by a proper nomenclature, possibly by having names from a pr-defined name space.

We can handle Class Constraints in a number of different ways. To deal with constraints representing unique-ness, we can use the built-in facilities of XML Schema such as unique, key, keyref etc. But general class constraints become constraint methods of the collection class associated with the user-defined class. Using the same concept as mentioned earlier 4.1(ii), we can describe class constraints “uniqueC1” which represents “id of an employee must be unique” in “Employee_Collection” class**.** As before, a new attribute “BC1” of type “uniqueC1” is added in the schema for the class “Employee-Collection”.

5. Interoperability between XML and Java Object Model

Interoperability between XML and Java Object Model requires the tool that transform the XML data from XML Schema representation to Java representation and vice versa. In this context, data binding is the process of mapping the components of a given data, such as SQL table or an XML schema, into a specific representation for a given programming language that depicts the intended meaning of the data format (such as objects, for example). XML Data binding consists of three primary concepts – the specification of XML to Object model bindings, marshalling and unmarshalling. Different binding frameworks will specify bindings differently and typically a data binding framework will have more than one way to specify such bindings. A binding schema specifies details about how classes are generated from XML schema and vice versa. A binding schema also provides facility to specify type conversions, name transformations and specification of super classes for generated objects.

In this context, Castor [8] has three core data binding methodologies: automatic data binding using built-in introspection, user specified data binding and XML schema based data binding with complete source code generation. These methodologies can be used independently or together, which makes Castor a very powerful and easy to use Data Binding Framework. The motivation behind using Castor is powerfulness, simplicity and supporting conversion from XML Schema to Java Programming Language. During translation from XML Schema to Java classes, the Castor Data Binding package generates an associated class definition for each class definition created, called Class Descriptor class. The Class Descriptors are classes that hold the binding and validation information for their associated class and used by the marshalling framework. During transformation for XML to Java, the Class Descriptors also contains the binding and validation information regarding constraints of XML data that are represented as attributes in XML schema.

In this section, we introduced an efficient way of storing XML data with constraint information into Java Object Oriented Programming Language. We have described two approaches to achieve such interoperability. We proposed an extension of the binding file of Castor Source Code Generator and the Castor XML Mapping file of Castor data binding tool to store the appropriate binding and validation information regarding constraints into class descriptor classes. To implement this, extension of some of the interfaces and classes of Castor data binding tool are also identified and completely specified. In the second approach, we have developed a wrapper-based translator, which can be used for interoperability between XML data with constraints into Java Source Code in addition with Castor Data Binging Tool.

5.1. Extension of the Binding File of the Castor Source Code Generator

The aim of this section is to provide an extension of the binding file of the Castor Source Code Generator for taking care about constraints during translation from XML Schema to Java Source Code. We introduced the extension of the componentBinding element of the binding file is as follows.

<componentbinding> element
<elementBinding|attributeBinding|complexTypeBinding|groupBinding name = xsd:string>
((java-class|interface|member|**constraint**),elementBinding*, attributeBinding*, complexTypeBinding*, groupBinding)
</elementBinding|attributeBinding|complexTypeBinding|groupBinding>

Here, the first child element (<java-class/>, <interface>, <member> or <constraint> will determine the type of binding. The following is the brief description of newly introduced element **constraint** of the binding file of the Castor Source Code Generator.

<constraint> element

<constraint	name? = xsd:string	ctype? = xsd:schemaType
	return-type? = xsd:Boolean	constraint-code? = xsd:string
	constraint-type? = xsd:string	attribute-list? = xsd:string
	handler? = xsd:string	validator? = xsd:string />

The detailed description of the attributes of newly introduced element constraint is as follows.

name – the name of the constraint that will be generated.
ctype – the SchemaType that corresponds to the java type chosen to represent the constraint
return-type– the return type of the constraint method which is boolean.
constraint-code – the body code for the constraint method.
constraint-type – represents the type of the constraint, whether it is a constrained attribute or constrained method or class constraint.
attribute-list – represents the attributes associated with the constraint.
handler - the fully qualified name of the constraintHandler to use.
validator - the fully qualified name of the constraintValidator to use.

5.2 Extension of the Castor XML Mapping File

The objective of this section is to extend the Castor XML Mapping file, so that the Class Descriptors classes contains the corresponding binding and validation information regarding integrity constraints during two-way interoperability between XML and Java. Castor XML Mapping is a good way to simplify the binding of Java classes to XML document. Castor allows one to specify some of its marshalling/unmarshalling behavior using a mapping file outside the Castor's default behavior.

(i) First, we have introduced the extension of the class element of Castor XML mapping file, which contains all the information used to map a Java class into an XML document is as below.

<class> element
<! ELEMENT class (description?, cache-type?, Map-to?, field+, **constraint+**)
<! ATTLIST class……………. >
where **constraint** represents zero or more <**constraint**> to describe the constraint information associated with the Java objects.

(ii) A new element <constraint> is introduced into Castor XML mapping file, which is as below.

<constraint> element
<! ELEMENT constraint (description?, bind-xmlconst?)>

<! ATTLIST constraint	name NMTOKEN #REQUIRED	
	ctype NMTOKEN #IMPLIED	
	constraint-type(cAttribute | cMethod | cClass)	return-type(boolean)
	handler NMTOKEN #IMPLIED	direct(true | false) "true"
	conscreate-method NMTOKEN #IMPLIED >	

The following is the brief description of the elements and the attributes of the <constraint> element.

description	–	an optional <description>.
bind-xmlconst	–	the name of the XML element or attribute which will be mapped into java constraint.
name	–	the java constraint name.
ctype	–	the SchemaType that corresponds to the java type chosen to represent the constraint.
return-type	–	the return type of the constraint method which is Boolean by default.
constraint-type	–	represents the type of the constraint, whether it is a constrained attribute or constrained method or class constraint.
handler	-	the fully qualified name of the constraint handler to use.

direct - If true, Castor will expect a public constraint method in the object. By default, this is true because all constraint methods should be public.
conscreate-method- Method for the creation or instantiation of constraint.

(iii) The element <**bind-xmlconst**> will be used to describe how a given java constraint should appear in an XML document. It is used both for marshalling and unmarshalling. The element is described below.

<bind-xmlconst> element

```
<! ELEMENT bind-xmlconst EMPTY>
<! ATTLIST    bind-xmlconst
              name NMTOKEN#IMPLIED          type NMTOKEN #IMPLIED
              location CDATA #IMPLIED       matches NMTOKENS #IMPLIED
              node    ( attribute | element | text ) />
```

The following is the detailed description of the attributes of the <bind-xmlconst> element.
name – the name of the element or attribute of XML Schema.
type - the name of the XML Schema type that requires for specific handling in the Castor Marshalling Frame Work.
location - allows the user to specify the sub-path for which the value should be marshaled to and from.
matches - allows overriding the matches rules for the name of the element. It is a standard regular expression and will be used instead of name attribute. For example, we can use the constraint names as C* or BC*.

To implement this, the brief description of extension of some of the interfaces and classes of Castor data binding tool are identified and completely specified in [15].

5.3. Brief Description of the Wrapper-based translator

A translator serves as an inter-mediator between different systems. The translation function is anticipated by implemented as part of a software wrapper. In this section, we introduced a wrapper to convert the Java Source Code produced by Castor (suppose representation A) into our presentable Java Source Code format (suppose representation B).

A wrapper is a piece of software used to alter the view provided by one interface to another without modifying the underlying component code. The translator must be capable of converting instances of a class's attributes, methods and constraints (or both attributes, methods and constraints in the form of an object of the class) from representation A to representation B and vice versa. Since we are only concerned about the classes with constraint information, the operational parameters can either be attributes, constraints, methods, objects or their combinations. The wrapper would intercept the parameters and follow the appropriate translation rule to accomplish conversion from representation A to representation B and vice versa. A set of translation rules govern the translations that take place in the wrapper is as follows.

Rule1. For every class definition C (either general purpose or collection class) in representation A, a corresponding class C' is created in representation B. Similarly, for every class definition C' in representation B, a corresponding class C is created in representation A.
Rule2. For every field definition F whose names are not like –C* or _BC* in representation A, a corresponding field F' is created and added to the corresponding class definition in representation B. Similarly, for every field definition F' in representation B, a corresponding field F is created and added to the corresponding class definition in representation A.
Rule3. For every Method definition M in representation A, a corresponding method M' is created and added to the corresponding class definition in representation B. For every Method definition M' in representation B, a corresponding method M is created and added to the corresponding class definition in representation A.
Rule4. For every field definition F whose names are like –C* or _BC* in representation A, a corresponding constraint Boolean method is created and added to the corresponding class definition in representation B. Similarly,

for every constraint Boolean Method in representation B, a corresponding field F is created and added to the corresponding class definition in representation A.

The wrapper would invoke the appropriate translation rule to convert the Java Source code from representation A to representation B and also from representation B to representation A. Now, the resultant structure of the wrapper class is as below.

```
Public class wrapper
{
private String  className;
private String fieldname[]=new String[50];
private String methodName[]=new String[50];
private int  totalNoofFields;
private int totalNoofMethods;
private int totalNoofConstraints;
public void getClassName (BufferedReader br ) throws IOException;
public void getFieldName (BufferedReader br ) throws IOException;
public void getMethodName (BufferedReader br ) throws IOException;
public void getFieldType (BufferedReader br, FileWriter fw1) throws IOException;
public void createClass (BufferedReader br, FileWriter fw) throws IOException;
public void fieldToField (FileWriter fw) throws IOException;
public void methodToMethod (FileWriter fw) throws IOException;
public void fieldToConst (BufferedReader br, FileWriter fw) throws IOException;
public void constToField (BufferedReader br, FileWriter fw) throws IOException;
}
```

The brief description of the methods of the wrapper classes is as follows:

public void getClassName(BufferedReader br) throws IOException - This method is used for getting the name of the class.
public void getFieldName(BufferedReader br) throws IOException – This method is used for getting the name of the field.
public void getMethodName(BufferedReader br) throws IOException – This method is used for getting the name of the method.
public void getFieldType(BufferedReader br, FileWriter fw) throws IOException - This method is used for getting the corresponding field type.
public void createClass(BufferedReader br, FileWriter fw) throws IOException – This method is used for create a class definition from representation A to representation B and vice versa.
public void fieldToField(FileWriter fw) throws IOException – This method is used for create a field definition from representation A to representation B and vice versa and add to the corresponding class definition.
public void methodToMethod(FileWriter fw) throws IOException – This method is used for create a method definition from representation A to representation B and vice versa and add to the corresponding class definition.
public void fieldToConst(BufferedReader br, FileWriter fw) throws IOException - This method is used for create a Constraint method definition in representation B from the field definition of representation A and add to the corresponding class definition.
public void constToField(BufferedReader br, FileWriter fw) throws IOException - This method is used for create a field definition in representation B from a constraint method definition of representation B and add to the corresponding class definition.

For translations involving more than one class definitions, the process must be repeated to translate all the classes of representation A to representation B and vice versa.

5.4 An Example of Java Source Code generated by Wrapper

The XML Schema "Employee" with constraint information presented in Section 4.1.(ii) will be mapped into the following Java source code with the help of Wrapper in addition with extended Castor Data Binding Tool.

```
Public Class Emp
        {Public idC   Idno;           Public String Name;
         Public cage Age;             Public String Designation;
         Public Float Salary;         Public    Byte      Experience;
         Public  cStatus   Status;

        Public    Boolean C1()
        {If ((this.Experience <=5) && (this.Salary > 2500))
        return false;
        else
        return true; } }
```

6. Conclusion

We have introduced a constraint model for Object Oriented data. We have further formalized the model by expressing it using UML and ODL. To increase interoperability, we have described a scheme for representing XML data with constraint information using XML Schema definition Language. In order to interact with data into a more manageable programming language format, a wrapper based translator have been developed to accomplish two-way interoperability between XML and Java object Model. Middle wares such as CORBA, EJB are developed to accommodate distributed objects and to facilitate efficient manipulation of these objects. Therefore, for efficient manipulation of distributed data and objects with constraint information, another important direction of extension of the constraint model in objects distributed over a network maintained by a middle wares such as CORBA or EJB.

References

1. Bassiiliades, N., Vlahavas, I., “Modeling constraints with exceptions in object oriented Database”, Proceedings of 13th international Conference on the Entity- Relationship Approach, Manchester, United Kingdom, December 1994, Lecture Notes in Computer Science, P. Loucopoulos (Ed.), Vol. 881,1994, 189-201.
2. Bagui, S., “Achievements and Weaknesses of Object-Oriented databases”, Journal of Object Technology, July/August 2003, Vol.2, No. 4, 29-41.
3. Bohannon, P., Freire, J., Roy, P., Siméon, J., “From XML Schema To Relations: A Cost-Based Approach to XML Storage”, Proceedings of the 18th International Conference on Data Engineering (ICDE'02), SanJose, California, February 26 – March 1, 2002.
4. Cuppens, F., Gabillon, A., “Rules for Designing Multilevel Object-Oriented Databases” Proceedings of 5th European Symposium on Research in Computer Security (ESORICS 98), 1998, 159-174.
5. Formica, A., “Finite Satisfiability of Integrity Constraints in Object-Oriented Database Schemas”, IEEE Transaction on Knowledge and Data Engineering, January/February 2002 (Vol. 14, No. 1), 123-139.
6. Fan, W., Simeon, J., “Integrity Constraints for XML”, Proceedings of ACM Symposium on Principles of Database Systems (PODS), Dallas, Texas May, 2000, 23-24.
7. Fan, W., Simeon, J., Kuper, G. M., “A Unified Constraint Model for XML”, Proceedings of ACM SIGMOD conference on Management of Data, HongKong, May 1-5, 2001.
8. Castor Data Binding Framework, http://www.castor.org.
9. Hwang, S., Lee, S., “Modeling Semantic Relationships and Constraints in Object-Oriented Databases, Proceedings of the 1990 ACM SIGBDP Conference on Trends and Directions in Expert Systems, Orlando, Florida, US, 1990, 396-416.
10. OMG Unified Modeling Language, Version 1.5 Available: http://www.omg.org.
11. Ray, C., Chattopadhyay, S., Bhattacharya, S., “Modeling Constraints as Methods in Object Oriented Data Model” Proceedings of the International workshop on Engineering Automation for Software Intensive System Integration, Monterey, California, USA, June18-22, 2001, 220-227.
12. Ray, C., Chattopadhyay, S., Chattopadhyay, M., “Expressing Constraint Models in Object Oriented Data Using UML and ODL”, International Workshop on Distributed Computing, Calcutta, India, December 28-31, 2002,LNCS#2571, Springer Verlag, 329-338.

13. Ray, C., Chattopadhyay, S., Bhattacharya, S., “Representation of Constraints in Object Oriented Data Using XML”, International Workshop on Advanced Computing (ADCOM 2003), Coimbatore, India, December 17-23, 2003.
14. Ray, C., Chattopadhyay, S., Bhattacharya, S., “Transforming XML Data with Constraint Information into Java Object Model”, International Workshop on Advanced Computing (ADCOM 2004), Ahmedabad, India, December 15-18, 2004, 751-758.
15. Widener, P., Schwan, K., Eisenhauer, G., “Open Metadata formats: Efficient XML-Based Communication for Heterogeneous Distributed Systems”, Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS’01’), January, 2001.
16. Yang, Y., Choi, H.S., Lee, S., Lee, Su., “XML-based Retrieval of Object–Oriented Frameworks”, IEEE Transaction on Knowledge and data Engineering, 2000.
17. Slack, J.M., Unger, E.A., “A Model of Integrity for Object-oriented Database Systems, Proceedings of the ACM/SIGAPP Symposium on Applied Computing technological challenges of the 1990’s”, Kansas City, Missouri, USA, 1992, ISBN 0-89791-502-X, 234-242.
18. Young, P., Berzins, V., Luqi,J. G., “Use of Object Oriented Model for Interoperability in Wrapper-Based Translator for Resolving Representational Differences between Heterogeneous Systems”, International workshop on ‘Engineering Automation for Software Intensive System integration’, Monterey, California, USA, June18-22, 2001.

A Framework towards Web Service Composition Modeling and Execution

Muhammad Adeel Talib, Zongkai Yang

Electronic & Information Engineering Department,
Huazhong University of Science & Technology,
Wuhan, Peoples Republic of China
matalib@gmail.com; zkyang@public.wh.hb.cn

Abstract. The biggest challenge of business process management is the provision of non-technical tools, based on implementation standards, which swing control of business processes away from technical departments and towards the business process owners themselves. These tools aid business users in designing high level process models using graphical notations which can then be mapped to lower level implementation models for execution. In this paper we propose our framework leading to a tool that aids business user in designing Web service based process (or in other words, Web service compositions) in BPEL4WS. We elaborate what information is required from the user in order to model the composition and how the technological details can be hidden from her. It is our conjecture that such tool will facilitate Web service composition design and development by giving an upper hand to business users – the people who actually conceptualize the processes.

1 Introduction

Used together, business process management and service-oriented architecture can form a dynamic combination that leverages the agility and extends the capabilities of both technologies. In this arena individual Web services are federated into composite services with value added functionality. Organizations can encapsulate their business functions as Web services and create virtual processes that interact with other organizations' processes. The interaction logic is specified as an XML based business process language. One such candidate language that seems to have attracted the most attention at the moment is the Business Process Execution Language for Web Services (BPEL4WS or BPEL in short) [1] that was originally drafted by BEA, IBM and Microsoft and which is now being formalized by a committee at OASIS. Though the language is still going through refinement with its new version under planning phase, it has gained a lot of attraction and support from the industry and has become the *de facto* standard [2].

According to the BPEL specification, BPEL defines a model for describing the behavior of a business process based on interactions between the process and its partners. This statement creates a misconception that BPEL is a business process modeling language. It is rather an execution language [business process modeling and standardization]. Like other XML based languages, it is of textual form and contains complex constructs and not so easy semantics. Business operations people are used to flow diagrams and other graphical notations instead of textual notations. BPEL attempts to offer the best by introducing a flow construct and using links to create 'arbitrary' flow dependencies between the activities contained within the flow construct. However, the semantics relies on a complicated formulation which tests and propagates the status of links. This makes it difficult for a business user, who actually conceptualizes the process, to model the process in it. There is a need to develop a methodology in order to assist the business analyst, who is not a technology expert, to model process compositions. Our aim is to develop a tool that captures the explicitly required information about a composition from the business modeler while at the same time hiding the technological details from her.

In this paper, we have briefly describe our proposed framework that can capture high level process composition requirements in an abstract way and then automatically transforms the high level process design model into low level process execution model i.e. BPEL. We identify various concepts of BPEL metamodel and see how these can be captured with abstraction from the user being a major concern.

Rest of the paper is organized as follows: First of all we present the related work in Section 2 as it gives an idea about the significance of our work. In Section 3 we list various concepts which are required to be captured in order to produce a Web service composition. Then in Section 4 we briefly discuss our proposed framework and describe how it facilitates the business user in capturing the BPEL semantics. Finally we conclude in Section 5 along with future work.

2 Related Work

BPEL as well as other process composition languages are of textual form and the specifications written in them are difficult to understand and visualize [3]. Towards this end various graphical specifications have been proposed to model Web service compositions that sit on top of BPEL in the technology stack. These tend to reduce the design complexity by provided graphical representations. Authors in [4] describe a UML profile and transformation rules that can be used to produce UML models of Web service compositions. The operation signatures are modeled as UML class model and the behavior is modeled in UML activity diagram thus producing a new service model. [5] present a similar UML profile and mapping rules to BPEL. Although these UML based modeling techniques provide a helping hand to developers to model applications, they do not abstract away the syntactic details from the user. The user still has to acquire sound knowledge of the underlying specifications as well as the graphical notations along with their usage scenarios. Besides, as quoted in [6], the gap between UML and BPEL is very large which makes the mapping quite complex.

BPMI has recently proposed a language – Business Process Management Notation (BPMN) [7] which provides graphical constructs with mapping to BPEL. Numerous commercial BPEL implementations are available where the vendors provide visual designing tools such as BPWS4J from IBM and BPEL Server from Oracle. In both cases either the user is bound to learn a new graphical language or should have sound knowledge of BPEL constructs. [8] present an approach to visually model Web services composition using Object-Process Diagrams (OPD). The paper describes a two-way transformation from OPD to BPEL using OPD templates. Again, the approach assumes the user is familiar with BPEL and OPDs. In contrast to UML and other graphical modeling based solutions, we focus on relieving the user from the syntactic details of BPEL along with the modeling notation and providing her a user friendly graphical interface where she can design the composition in step by step manner without any prerequisite modeling or programming skills.

Authors in [9] have proposed a template based web service composition model for automatic code generation of business process languages. The degree to which automatization of service composition is achieved depends on the availability of templates (to the designer) that fit into the desired composition pattern. The availability of templates, in turn, depends on access to public registries for retrieval. Currently there is no such provision and is in the future scope of the project. Our conjecture is that these templates can provide automatic code generation only for a part of code that is repeatable but, cannot be used to generate the whole process flow, for that the designer has to gain knowledge of complex design patterns. Our framework focuses on the whole process flow design instead of partial code generation.

3 Concepts of BPEL metamodel

The full explanation of BPEL semantics is out of the scope of this paper. Here we only discuss the concepts involved. The metamodel of BPEL incorporate the following concepts [10] that represent the operational, behavioral, informational, organizational and transactional aspects of the language:

1. *Task I/O:* Task refer to basic units of work or activity. The input and output (I/O) of these tasks may be modeled using simple or XML complex types.
2. *Task Address:* The address specifies where or how a service can be located to perform a task. The address can be modeled directly via a URI reference of a service or indirectly via a query that identifies a service address.
3. *Control Flow:* The control flow defines the temporal and logical relationships between different tasks. Control flow can be specified via directed graphs or block oriented nesting of control instructions.
4. *Data Handling:* Data handling specifies which variables are used in a process instance and how the actual values of these variables are calculated.
5. *Instance Identity:* This concept addresses how a process instance and related messages are identified. Correlation uses a set of message elements that are unique for a process instance in order to route messages to process instances.
6. *Roles:* Roles provide for an abstraction of participants in a process.

7. *Events:* Events represent real-world changes. Respective event handlers provide the means to respond to them in a predefined way.
8. *Exceptions:* Exceptions or faults describe errors during the execution of a process. In case of exceptions dedicated exception handlers undo unsuccessful tasks or terminate the process instance.
9. *Transactions:* Business transactions represent long-running transactions. In case of failure the effects of a business transaction are erased by a compensation process.

Capturing these concepts of BPEL while at the same time hiding the syntactic details is not an easy task. The higher level metamodel required to do so must be business user friendly that captures information relevant to much lower level BPEL implementation model. This difficultly also implies to the automatic transformation between the two metamodels. We advocate the use of a user friendly interactive graphical interface instead of a graphical modeling language to capture these concepts. The information is stored in a relational model (Fig. 1) which is then transformed into the BPEL model automatically.

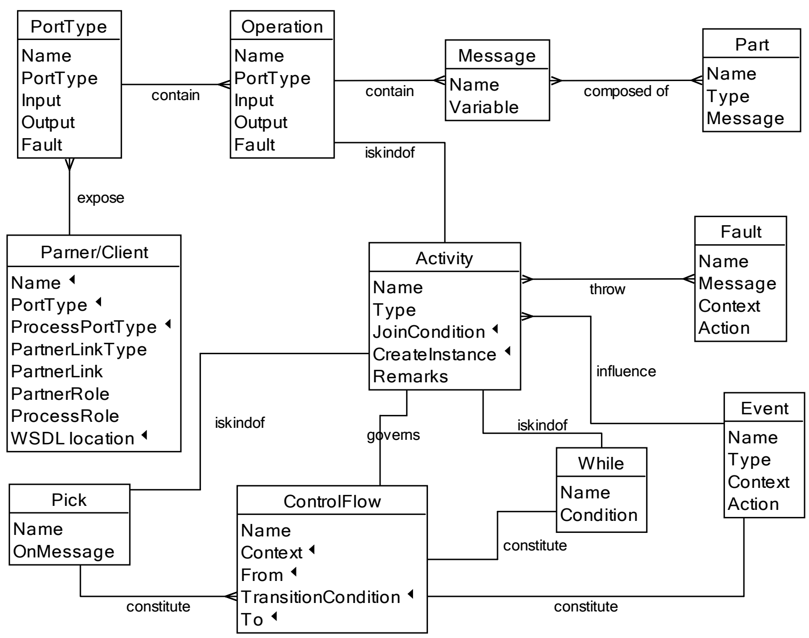


Fig. 1. Relational model which stores the composition information. The marked attributes represent the only information provided by the user

It should be noted that only basic understanding of the concepts of BPEL is required by the user and not its syntactic details. Hiding syntactic details from the business user is the basic theme of our research. Our goal is to provide composition modeler the flexibility to express less than complete detail without prerequisite high specification knowledge. Refinement is then a natural process of adding further detail, while still conforming to the laws of composition. The motive here is to facilitate the business user, who does not know much detail about composition language, to construct.

4 Proposed Framework

Through the Graphical User Interface (GUI), the system captures the information required to develop the composition from the Composition Modeler and stores it in a Relational Repository. Knowledge about the parties involved in the collaborations, their interface description file locations, order of activities, instance creation, correlation token, etc. is captured in an incremental fashion in such a way that the modeler remains unaware of the underlying syntax. The Inference Engine automatically infers additional information from the user provided information using inference algorithm (an algorithm that takes values of attributes from the relational model and deduces other attributes). Once all information is captured, transformation rules are applied to map from relational to BPEL metamodel/schema (the word schema is more appropriate to use here as both the relational and BPEL model use schemas as their interchange format). This is done by the Transformation Engine. Once the code is generated, it must be validated for accuracy and verified against deadlocks. This is done using an existing BPEL Validator. The final validated and verified BPEL file and partner interfaces are passed on to an execution engine that executes the composition. The architectural components of our proposed framework are shown in Figure 2.

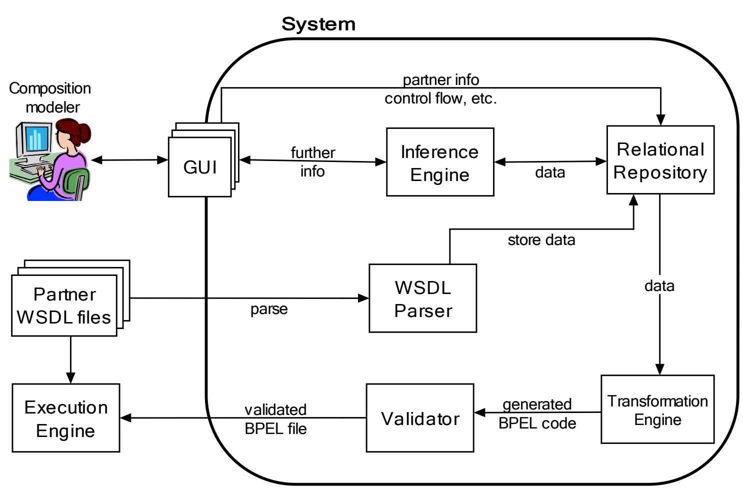


Fig. 2. Conceptual architecture of proposed framework

4.1 Information Capturing

This section describes briefly how our framework captures the BPEL concepts (given in Section 3) keeping the user unaware of the syntactic detail and what information can be inferred from the inference algorithm.

4.1.1 Task I/O. This information is deduced from WSDL Parser. It parses the partner interface files provided by the user and deduces the port types exposed by the collaborating parties, the operations offered, the messages involved and the parts of the messages that correspond to basic or complex data types. Fig. 3 shows mapping between the partner WSDL file and the relational model.

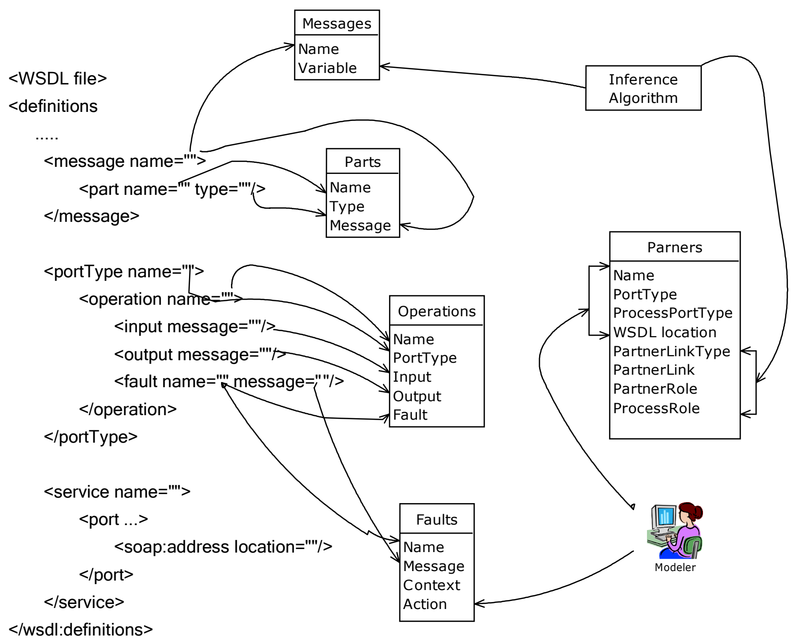


Fig. 3. Mapping from WSDL files to relational model done by WSDL parser. The figure also shows what information is provided by the user and what is inferred from the inference algo.

4.1.2. Task Address. The task address is a URL and is also extracted from the WSDL files from binding element.

4.1.3. Control Flow. The modeler captures the control logic with the help of simple control rules that identify which activity has to be executed after an activity is completed under what conditions. Structure of a control rule is as shown below:

```
ControlRule {
activity (messaging|basic|while|pick|event)
postActivity (messaging|basic|while|pick|event)
transitionCondition (Boolean expression)
}
```

Synchronization (interdependencies) among the activities is handled by BPEL link semantics which mark the parent node as source activity and the child node as target activity. If the target activity is guarded by a condition, the source activity will specify the transition condition. Based on this control link semantics of BPEL, the transformation rules synchronize the activities in a single flow. We believe that this graph based technique allows more abstraction as compared to structured formation of BPEL specification, in that the modeler has to explicitly define the complex structure of composition.

Here inference algorithm is used to decide whether a messaging activities i.e. operation is associated with a receive, reply or invoke construct. The modeler just defines the order in which the activities have to be run without knowing when to use the BPEL messaging constructs. The use of control rules also facilitates the business user in updating changes in the business logic without relying on the developer, providing agility against change.

4.1.4. Data Handling. The message exchange variables are deduced from the description files of the partner interfaces. For each unique message taking part in the collaboration, there is a unique variable. Intermediate variables used to store data during business logic manipulation have to be defined by the user.

4.1.5. Instance Identity. Three instantiation patterns are involved during instance creation: Single start; Multiple start with receive; and Multiple start with pick. The modeler has to select the identified pattern according to the business logic. Correlations are usually context-dependent and thus cannot be derived by general rules. They have to be defined by the user. We are currently working how to abstract the correlation constructs from the user.

4.1.6. Roles. In a business collaboration modeled by BPEL, there is a centralized coordinating authority, the process, which interacts with other parties (i.e. partner and client). It is important to distinguish between a partner and a client. A partner is the party that provides services to the process. Client on the other hand gets service from the process. It is required from the modeler only to provide the port types exposed by the partner or client. The partnerLinkType, partnerLink, partnerRole and myRole constructs can be deduced from the inference algorithm. The detail of the inference algorithm is out of the scope of this paper but a snippet of the algorithm inferring the role semantics of a partner is listed below:

```
for each partner involved {
PartnerLinkType = Partner.Name + 'LinkType'
PartnerLink = Partner.Name + 'Link'
if (Partner.PortType != null then)
PartnerRole = Partner.Name + 'Provider'
if (Partner.ProcessPortType != null then)
MyRole = Partner.Name + 'Requester'
}
```

4.1.7. Events. Events represent real-world changes and their occurrence is captured by the control rules. The activity which is influenced by an event is encapsulated in a scope having an event handler attached that executes the actions to be performed. The actions to be performed are again captured with the help of control rules. The flow thus created will be nested inside the event activity.

4.1.8. Exceptions. Our framework automatically captures the application exceptions from the WSDL files and attaches fault handlers with the affected activities. Handling of system exceptions and compensation is yet to be explored.

5 Conclusions and Future Work

In this paper we have presented our idea for generating BPEL code for static composition semi-automatically. First we described various aspects of BPEL to be captured while modeling the business process choreography. We discussed what information can be abstracted from the user and what cannot. Then we proposed a relational model to store the captured information which is later transformed into BPEL process specification model. We also briefly described how the relational model is populated with information plugged in by the modeler and information derived by inference algorithms. It is our thesis that using this approach a business user with limited BPEL knowledge can have the capability to design complex business to business Web service compositions.

In order to verify our concept a tool is under development. For the sake of simplicity we have abstracted from the namespace issues which are an integral part of BPEL specification. Both relational and process models have strong formal grounds (relational and process algebra). The possibility of formally transforming the relational model into a process model is yet to be explored. Currently our framework only generates abstract composition without considering concrete bindings with the partners, which will be a part of our future work. For now we concentrate on static compositions where the modeler provides the service location information. Our framework can be modified to handle dynamic compositions where the services are selected at runtime. For this functionality we propose the use of a service broker that provides the service URLs upon queries based on functional and non-functional requirements.

References

1. Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Version 1.1. Specification. BEA Systems, IBM Corp., Microsoft Corp., SAP AG, Siebel Systems. May, 2003. http://www.ibm.com/developerworks/ webservices/library/ws-bpel/
2. Harmon, P.: BPM Tools. Business Process Trends Newsletter, vol. 2, no. 4, Apr, 2004. http://www.bptrends.com
3. Wil M.P. van der Aalst: Web service composition languages: Old wine in new bottles? In: Proceedings of the 29th IEEE EUROMICRO Conference, pp. 298-307, Belek-Antalya, Turkey, Sep, 2003, 298-307
4. Skogan, D., Gronmo, R., Solheim, I.: Web Service Composition in UML. In: Proceeding of the 8th International IEEE Enterprise Distributed Object Computing Conference (EDOC'04), California, USA, Sep, 2004, 47-57
5. Gardner, T.: UML Modeling of Automated Business Processes with a mapping to BPEL4WS. In: Proceedings of 1st European Workshop on Object Orientation and Web Services (EOOWS'03), Darmstadt, Germany, 2003
6. Bézivin, J., Hammoudi, S., Lopes, D., Joualt, F.: Applying MDA approach to B2B applications: A road map. Workshop on Model Driven Development (WMDD 2004) at ECOOP 2004, Oslo, Norway, Vol. 3344 of Lecture Notes in Computer Science (LNCS), Springer-Verlag, Jun, 2004
7. White, S. A.: Business Process Modeling Notation. BPMN 1.0, Business Process Modeling Initiative. 2004. http://www.bpmn.org
8. Yin, L., Wenyin, L., Changjun, J.: Object-Process Diagrams as Explicit Graphic Tool for Web Service Composition. Journal of Integrated Design & Process Science: Transactions of the SDPS, Vol. 8, No. 1, 2004, 113-127
9. Karastoyanova, D., Buchmann, A.: A Procedure for Development and Execution of Process-based Composite Web Services. In: Proceedings of the IEEE International Conference on Web Engineering (ICWE'04), Munich, Germany, Jul 2004, 593-594
10. Mendling, J., Neumann, G.,Nüttgens, M.: A Comparison of XML Interchange Formats for Business Process Modelling. In: Proceedings of EMISA 2004, Luxembourg, In: Feltz, F., Oberweis, A., Otjacques, B., (eds.), Vol. 56 of Lecture Notes in Informatics (LNI), Oct, 2004, 129-140