Informatics Research Proposal: Mapping Fundamental Business Process Modelling Language to a Semantic Web based Language

Gayathri Nadarajan (s0450937)
Supervisor: Dr Jessica Chen-Burger
March 18, 2005

Overview
The need for more sophisticated web-based support tools has become apparent with the evolution of the World Wide Web and the Semantic Web vision\(^1\). For organisations with business goals, the automation of business processes as web services is increasingly important, especially with many business transactions taking place within the web today. With the existence of Business Process Modelling (BPM) methods and Workflow technologies, web services can benefit from these mature approaches to allow for the manipulation and execution of processes. Some effort have been made to overcome this pitfall [Chen-Burger, Stader 2003],[Guo et al 2004],[BPEL4WS 2005], yet there still exists a gap that needs to be addressed between enterprise modelling methods and web services. In this proposal we seek to find means to bridge the gap that exists between BPM methods and web services by attempting to map a semantic based process modelling language, Fundamental Business Process Modelling Language (FBPML)\(^2\) to a Semantic Web compliant language. A successful mapping would allow for provision of static and dynamic web services.

1 Statement of Problem - The Gap

Web services are trendy amongst many business organisations today with the pervasiveness of the World Wide Web and the advent in Web technology. Business-to-Business (B2B) Electronic Com-

\(^1\)Berners-Lee, Hendler, Lassila Scientific American, 2001
\(^2\)AIAI, University of Edinburgh, U.K.

merce is fast becoming the most important application area of Semantic Web technology in terms of market volume [Fensel et al. 2003]. Thus there is a pressing need to bring both business and technology together for the realisation of virtual organisations.

However, a main obstacle in bridging enterprise modelling methods and web services is the lack of direct mapping from EM methods to web services. Despite some on-going effort, there still exists a gap that needs to be addressed between enterprise modelling methods and web services. We propose a mapping from a semantic based Business Process Model, FBPML to a Semantic Web compliant language in order to narrow the gap.

2 Background

Enterprise modelling (EM) methods have been recognised for their value in providing a more organised way to describe complex, informal domain. Many EM methods have emerged to describe and redesign businesses, namely business process modelling, business system modelling and organisational context modelling methods [Chen-Burger, Robertson 2004]. Business Process Modelling (BPM) methods are able to formally express informally practised processes. More importantly, actions and effects of these processes can be demonstrated using simulation techniques. Some examples of business process modelling (BPM) methods representations include Process Specification Language (PSL), Integration Definition Language (IDEF0, IDEF1, IDEF1X and IDEF3), extended UML Activity Diagrams and Petri-Nets. More recently, new process languages have been developed
to promote the understanding and interoperability of process semantics over the Web, with the extensibility of operating over the Semantic Web. They are characterised by XML and XML-based languages, such as Resource Description Framework (RDF) and Web Ontology Language (OWL). Some of these languages include Business Process Execution Language for Web services (BPEL4WS), Business Process Modelling Language (BPML) and Web Service Ontology (OWL-S). For the purpose of this proposal, we make references to a semantic based BPM language, Fundamental Business Process Modelling Language (FBPML) and an ontology-based three-layered BPM approach incorporating FBPML to illustrate an example business process modeling method and workflow technology. Discussion on web services will be based on SOAP, WSDL and semantic web compliant languages (OWL-S and BPEL4WS).

2.1 Fundamental Business Process Modelling Language (FBPML)

FBPML [Chen-Burger et al. 2002] is a diagram-based business process modelling language which merges the visual capabilities of IDEF3 [IDEF3 1995] and the formal specification for semantics and theories provided by PSL [Schlenoff et al. 1997]. It is designed to support software and workflow system development. Therefore it can express business processes in conventional first order predicate logic [Kuo et al. 2004]. Figure 1 illustrates an example process model defined using FBPML.

2.1.1 Notations

A model described in FBPML is made up of Main Nodes, Junctions, Links and Annotations. The Main Nodes are summarised as follows:

- **Activity** is the main concept to denote a process which may be further decomposed or specialised into subprocesses. The three main components of an activity are trigger(s), precondition(s), and action(s).

- **Primitive Activity** is a leaf node activity that may not be further decomposed or specialised.

- **Role** is a specialised function, capability or authority possessed by an *enabler* over a set of activities.

- **Time Point** indicates a particular point in time during the process model.

**Links** between processes consist of **Precedence links** and **Synchronisation Bars** which place temporal constraints on process execution.

The four types of **Junctions** in FBPML are **Start**, **Finish**, **And** and **Or**. Junctions such as **And** and **Or** may be interpreted as joint or split nodes depending on the use in the diagram and may indicate one-to-many relationships with temporal constraints between the activities connected to them[Kuo et al]. **And_Joint** or **Or_Joint** indicates that there is more than one process preceding the **And** or **Or** junction but there is only one process following the junction. **And_Split** or **Or_Split** means that only one process will proceed to the **And** or **Or** junction, but more than one process will follow the junction. There may also be combination of **And** and **Or** junctions to represent a more complicated BPM.

**Annotations** include **Idea Note** and **Navigation Note**. Neither of them contribute to the formal semantics of the process model. Instead, they are used to help users to understand the processes more clearly from an intuitive point of view.

Sytanctically

![Figure 1: An example process described using FBPML notation](image)

*3A human or an autonomous agent*
2.2 Three-layered BPM Approach in a Workflow Engine

BPM methods have been closely related to workflow technologies in overcoming the inflexible and complicated nature of workflows. Workflow is a set of methods and technologies, which support a business process through the analysis, redesign and automation of information-based activities. A key aspect is coordination, which goes beyond simply communicating information through electronic media. It also involves individuals within an organisation, working on projects in order to implement prescribed business processes or achieve an overall business goal [Workflow 2000]. Essentially workflow involves the computerised facilitation or automation of business processes, in whole or part [WFMC 1995].

Initial workflow systems incorporating business processes did not separate business logic from implementation logic, making them brittle in their reaction to the dynamic and volatile environment within which they operate in. A major milestone was accomplished when a three-layered business process modelling approach was implemented and orchestrated onto a Workflow Management System (WFMS) in [Chen-Burger, Stader 2003] incorporating FBPML, which separated business and technical decisions while keeping track of design rationale for any technical decisions being made, thus giving WfMS greater flexibility and adaptability. This was assisted by the mapping of meta-ontologies between different domains.

2.3 Relevance to Web Services

A web service is a self-contained software system that can be published, located and invoked across the web by other applications [WS 2001]. The basic Web services stack is comprised of native Internet transport protocols such as Hyper Text Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), data models that are based on XML, messages exchanged in Simple Object Access Protocol (SOAP), the description of service operations and types via Web Services Description Language (WSDL), and Universal, Description, Discovery and Integration (UDDI) for publishing and discovery of services.

As stated in earlier sections, web services are becoming more and more ubiquitous and in demand. The relevance with BPM methods and workflow technologies is that web services are increasingly being described in terms of process models and realised in WfMS, hence creating a dependency and reliance for more versatile WfMS. Additionally, with the evolvement of the World Wide Web, Semantic Web technologies are becoming more and more influential, thus there will be a need to address these technologies and how BPM methods and Workflow technologies can be adapted to allow for the orchestration of Semantic Web services.

2.4 OWL-S

OWL-S [OWL-S 2004] is a Web service ontology written in the Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL).

OWL [McGuiness, van Harmelen 2004] is a W3C recommendation for a web ontology language and is the representation language for OWL-S ontologies. One reason OWL was chosen as the representation language for OWL-S is that it is compatible with XML and RDF [Klyne, Carroll 2004] while providing additional expressiveness thus allowing users to formally describe more types of classes, properties, individuals, and relationships than XML or RDF. OWL provides three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full. OWL Lite provides less expressive power than OWL DL and OWL Full and was not expressive

\[\text{formerly known as DARPA Agent Markup Language (DAML-S)}\]
enough for OWL-S needs. OWL DL provides maximum expressiveness while retaining computational completeness and decidability and thus is the choice implementers make when they are interested in efficient reasoning support.

The SWRL Semantic Web Rule Language [SWRL 2004] is a rule language that combines OWL with the rule markup language [RuleML 2004] providing a rule language compatible with OWL. SWRL has been submitted to the W3C as a potential starting point for a new rules working group that could produce a W3C semantic web rule language.

SWRL expressions may be used in OWL-S preconditions, process control conditions (such as if-then-else), and in effects expressions. SWRL expressions may also mention process inputs and outputs as variables thus linking the two languages together. The OWL-S use of SWRL remains in OWL DL (by quoting SWRL rules and thus considering them to be XML Literals). This connection with SWRL makes the OWL-S ontologies more powerful since it uses the expressive power of rules in a potential emerging standard. It also serves as an example of how one can use a rule language and stay within OWL DL, thus preserving efficient reasoning options.

OWL-S supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. OWL-S markup of Web services facilitates the automation of Web service tasks, including automated Web service discovery, execution, composition and interoperation. Following the layered approach to markup language development, the current version of OWL-S builds on OWL, which in turn, is built on top of the Resource Description Framework (RDF), which an XML-based data model. Figure 2 illustrates this layering approach.

WSDL, by default, specifies abstract types using XML Schema, whereas OWL-S allows for the definition of abstract types as (description logic-based) OWL classes.

OWL-S is becoming the de facto standard for the composition of Semantic Web Services. It is important to note that although this technology is relatively new, there have already been some strong criticisms against it; that it suffers conceptual ambiguity, lacks concise axiomitisation, designed too loosely and offers an overly narrow view on Web services [Mika et al. 2004].

2.5 Business Process Execution Language for Web Services (BPEL4WS)

In the more specific domain of business processes, BPEL4WS is a formal specification that models the behavior of Web services in business processes [BPEL4WS 2005]. By doing so, it extends the Web Services interaction model and enables it to support business transactions. It defines an interoperable integration model that should facilitate the expansion of automated process integration within corporations and in business-to-business spaces. BPEL4WS combines IBM’s Web Services Flow Language and Microsoft’s XLANG specifications, and replaces them.

BPEL4WS builds on and extends XML and Web Services specifications. It is expressed entirely in XML, uses and extends WSDL, and uses WSDL and XML Schema for the data model. The BPEL4WS process model is layered on top of several XML specifications: WSDL, XML Schema, and XPath. WSDL messages and XML Schema type definitions provide the data model used by BPEL4WS processes. XPath provides support for data manipulation. All external resources and partners are represented as WSDL services. Heavily influenced by WSDL, BPEL4WS provides extensi-
bility to accommodate future versions of these standards. In short, a BPEL4WS business process definition can be thought of as a template for creating business process instances.

2.6 Proposal for a Mapping Tool

Based on current findings on BPM methods and Web services, we propose two hypotheses to be tested. Firstly, to allow for the composition and execution of Semantic Web services utilising business process technologies, it would be most fitting to enable business process modelling languages to be translated to Semantic Web compliant languages. We propose to implement a mapping from FBPML to either OWL-S or BPEL4WS. An initial plan is to select OWL-S over BPEL4WS due to its structural similarity to FBPML. OWL-S is described by a data model and a process model, as is FBPML, whereas BPEL4WS doesn’t have an explicit data model, which may pose a difficulty. However, this may change in due course should we find OWL-S less appropriate for the task. We will then select BPEL4WS to continue our investigation in the project. A successful mapping would allow for the direct provision of semantically enabled web services, thus narrowing the gap between BPM methods and web services.

We also propose to look into the limitations of OWL-S and suggest ways to improve its structure and representation so that its expressive power would better suit FBPML, and BPM languages in general. However, given the time constraints, we shall not look into any implementation aspect of this task.

3 Methods and Workplan

Consider the task of projecting FBPML models to OWL-S as follows:

*Diagram-based process representations of a FBPML model are to be transformed to rules in first-order predicate logic before being transformed into OWL syntax (which is built on XML-based RDF), which can be then represented as a graph.*

This can be split into two major tasks; one being the mapping of the data models, and the other being the conceptual mapping of the process models.

3.1 Data Model Mapping

The data model mapping is essentially the translation of the FBPML Data Language (FBPML-DL) [Chen-Burger 2002] to OWL-S data model, described by OWL, which is layered on top of the XML-based RDF Schema data model. The task of this mapping can be seen as the implementation of a high-level compiler.

The FBPML-DL has a strong basis in logic, based on first-order predicate logic and set theory. The language is presented using predicates similar to those in Prolog. It has four parts:

- **Foundational Model** provides concepts, predicates and functions of background theories used in the language.

- **Core Data Language** introduces core predicates and functions for concepts that are common to many applications. Their semantic is defined using constructs from the Foundational Model.

- **Extension Data Language** includes predicates and functions that are additional to the Core Data Language and are often application and domain dependent.

- **Meta-predicates** may define axioms of an application model.

We propose a parser implemented in Prolog to read in these Prolog-like (first-ordered) predicates and translate them into machine-accessible syntax. The parser would read in FBPML-DL in first-order predicate logic and transform it into OWL syntax. This process could be attempted either directly from FBPML-DL to OWL, or in stages; from Prolog to XML, from XML to RDF/S and from RDF/S to OWL.

3.2 Process Model Mapping

Some effort has been channelled into the mapping of the FBPML Process Language (FBPML-PL) and the OWL-S process model [Guo et al. 2004]. The findings, however, did not produce a complete mapping procedure. Here, we discuss them briefly.
3.2.1 Comparison between FBPML and OWL-S

Some differences and similarities identified by [Guo et al. 2004] are:

- Their process model’s basic properties differ in their component names. These have been identified and mapped accordingly.
- FBPML provides a script for the implementation of a process model while OWL-S doesn’t.
- FBPML primary activity can be treated as the atomic process in OWL-S, since they are both undecomposable and both represent single step execution.

3.2.2 Mapping Obstacles

Some issues and problems related to the mapping between FBPML and OWL-S have been addressed as well. These include:

- Many FBPML models do not decompose hierarchically and these cause mapping problems.
- Generally, semantic web process models use programming-like control constructs as their basic building blocks which are inadequate for all the modelling issues.
- OWL-S process model also poses synchronisation problems, it can specify how to control several processes’ synchronisations within a single composite process but says nothing about the interaction of processes from different OWL-S process models. This is due to the fact that there are no concepts or variables within OWL-S for information passing and sharing.
- OWL-S process model does not provide a representation of the concept “Role” as defined in FBPML. “Role” allows us to group processes from different domains, a process can also be shared by different roles.
- From the point of extensibility, FBPML allows for greater extensibility by adding new roles, for example.
- There is also visualisation issues to be considered, such as how to represent OWL-S in the same level of expressiveness as FBPML.

The mapping is partial because of differences in control assumptions and process decompositions. We propose to investigate the reasons for the partial mapping solution and work on improving it.

3.3 Improving OWL-S

Although a relatively new and emerging technology, OWL-S has been criticised for its limitations and lack of expressive power. We anticipate that representation issues pertaining to OWL-S will need to be addressed for this purpose.

3.3.1 Expressive power of OWL-DL

The underlying logic of the OWL-S ontology is OWL Description Logic (OWL-DL) [Patel-Schneider et al. 2004], which is a subset of first-order logic. From the rules perspective, OWL-DL is restricted to tree-like rules, but provides both existentially and universally quantified variables and full, monotonic negation. From the description logic perspective, rules are restricted to universal quantification, but allow for the interaction of variables in arbitrary ways. We propose to look into combining OWL-DL and rules for describing Semantic Web ontologies to improve its expressivity. However, such a combination is prone to undecidability [Grosof et al. 2003], thus should be approached with care.

3.4 Workplan

The duration of the project is approximately 3 months (12 weeks). In this period, milestones and deliverables have been identified as follows (assuming the start date is 01/06/05):

<table>
<thead>
<tr>
<th>Date</th>
<th>Milestone</th>
<th>Deliverable(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15/06/05</td>
<td>Requirements,</td>
<td>Requirements document</td>
</tr>
<tr>
<td></td>
<td>(Week 2)</td>
<td></td>
</tr>
<tr>
<td>30/06/05</td>
<td>OWL-S Limitations</td>
<td>Draft Complete</td>
</tr>
<tr>
<td></td>
<td>(Week 4)</td>
<td></td>
</tr>
<tr>
<td>15/07/05</td>
<td>Data Model</td>
<td>Prolog Parser Complete</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Week 6)</td>
<td></td>
</tr>
<tr>
<td>31/07/05</td>
<td>Process Model</td>
<td>Prolog Translator</td>
</tr>
<tr>
<td></td>
<td>Implementation</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>(Week 9)</td>
<td></td>
</tr>
<tr>
<td>10/08/05</td>
<td>Evaluation</td>
<td>Test Results Produced</td>
</tr>
<tr>
<td></td>
<td>framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Week 10)</td>
<td></td>
</tr>
<tr>
<td>25/08/05</td>
<td>Write-Up</td>
<td>Dissertation Complete</td>
</tr>
<tr>
<td></td>
<td>(Week 12)</td>
<td></td>
</tr>
</tbody>
</table>
Remarks:
- An extended period of time is allocated for the requirements to change after the Requirements Document is produced. This is in line with the possibility of a contingency plan should the original implementation fall through.
- Any development work should not exceed Week 10, which is the first week of August.

4 Evaluation

Although constant testing will be conducted throughout the development of the mapping system, two major evaluations will be necessary to justify the hypotheses proposed. One is the evaluation the correctness of the translated OWL-S produced and the second is the evaluation of the capabilities of the BPM through a rigorous testing of their instantiations as OWL-S constructs.

4.1 Correctness of the Translated OWL-S

The translated OWL-S data models and process models are to be evaluated for correctness. A simple way of describing this is by extracting the process and its arguments that have resulted from the mapping procedure and to compare if these correspond to the original values contained within the FBPML. Among the parameters that are to be checked for correctness:
- Process name
- Temporal constraints (AND, OR, synchronised AND and OR, XOR(optional) )
- Values of variables and constants contained within the process.

4.2 Case Studies using the Mapping Tool

For this part of the evaluation, we create different representation types of FBPML models, map them to the corresponding OWL-S using our implemented parsers, and test to see if all constructs can be translated successfully. This is an evaluation of the range of capabilities and limitations possessed by the developed mapping tool. Instances, ranging from trivial to complicated FBPML models are constructed based on one or more domains in question and tested as inputs to the mapping tool. An example domain that could be adapted is the PC Configuration Application from the KBST-EM (Knowledge Based Support Tool for Enterprise Modelling) modelling system which describes its process models using FBPML. Evaluation should bring forth the types of BPM models that can and cannot be mapped appropriately to OWL-S or BPEL4WS.

5 Expected Outcomes

Both practical and theoretical outcomes are anticipated for this project. Two parsers implemented using Sicstus Prolog are to be developed for the mapping of the FBPML data models and process models to OWL-S. The evaluation of the correctness and range and capabilities of the Mapping Tool will result in an analysis of the degree of compatibility between BPMs and Semantic Web technologies. While the study on the limitations of OWL-S would result in a specification for the features to be improved, and a suggested methodology to overcome them.

6 Schedule

The work breakdown can be viewed in the following summarised timeline:

6 Conclusions & Future Directions

We have proposed two conjectures on bridging the gap between BPM methods and Semantic Web Services. The mapping of data model and process

\[ AIAI, CISA, Informatics, The University of Edinburgh \]
model will be the basis of our methodology. While the evaluation framework and a critical analysis on OWL-S would bring forth some of the issues pertaining to the integration of BPM methods onto Web services. An interesting direction to pursue from here is to look into related technologies to both BPM methods and Semantic Web technologies, such as open architectures (e.g. Grid Services). Another more fundamental issue is to explore means to enhance OWL-S’s expressivity via improving its representation. This is a timely effort as knowledge representation and reasoning is presently a key research area into realising the Semantic Web.

References

[BPEL4WS 2005]

[Chen-Burger 2002]
Yun-Heh Chen-Burger, Informal Semantics for the FBPML Data Language Informatics Report Series: EDI-INF-RR-0154, School of Informatics, the University of Edinburgh, October 2002.

[Chen-Burger,Stader 2003]

[Chen-Burger et. al 2003]

[Chen-Burger,Robertson 2005]

[Fensel et. al 2003]

[Grosof et. al 2003]

[Guo et. al 2004]

[IDEF3 1995]

[Klyne,Carroll 2004]

[Kuo et. al 2004]

[Mika et. al 2004]

[McGuiness,van Harmelen 2004]

[OWL-S 2004]
OWL-S 1.1 Release. http://www.daml.org/services/owl-s/1.1/

[Patel-Schneider et. al 2004]
Peter F. Patel-Schneider, Patrick Hayes, and Ian Horrocks, Editors. OWL Web Onto-

[Schlenoff et al. 1997]

[SWRL 2004]

[RuleML 2004]

[WFMC 1995]

[Workflow 2000]

[WS 2001]
Tracy Gradner, An Introduction to Web Services, Ariadne Issue 29, IBM Corp.