D1.3. Business Process Modeling Ontology BPMO final version

Emilia Cimpian[1], Srdjan Komazec[1], Daniel Lintner[2], Clemens Blamauer[2], Mark Evenson[3]

[1]*Digital Enterprise Research Institute*
Leopold − Franzens Universität Innsbruck
Austria
{firstname.lastname}@deri.org

[2]*Distributed Systems Group*
Information Systems Institute
Vienna University of Technology
Austria
{lastname}@infosys.tuwien.ac.at

[3]*eTel Austria GmbH Und Co KG*
{firstname.lastname}@etel.at

Abstract

The semantic representation of business processes is lifting the modeling and mangement of Business Processes over the next evolutionary step. The development of a conceptual model for semantic business processes requires a joint effort of business experts and academia. This document presents the final version of the Business Process Modeling Ontology developed as part of the SemBiz project, as a result of the collaboration between two research groups and two Austrian industrial partners.

Contents

1 Introduction

The Business Process Modeling Ontology (BPMO) described in this document presents an innovative approach for semantic representation of business processes. BPMO was previously presented in the SemBiz deliverables [10] and [9] and evaluated as part of the deliverable [2]. The final version of the ontology is based on the previous versions ([10] and [9]), as well as on the results of the ontology evaluation based on the OntoClean [5][4] methodology (as presented in [2]).

This deliverable consists of WSML files describing all the concepts and ontologies developed as part of the SemBiz project for semantically describing the business processes, and this accompanying document which has the purpose of explaining all the concepts defined in the ontologies. The technologies and specifications consulted in order to develop the BPMO were previously described in [10] and [9], and will not be presented again in this document.

This document is structured as follows: Section 2 presents the overview of the BPMO, an explanation of the WSML files accompanying this document. Section 3 defines the relationship between BPMO and WSMO, specifying the role played by WSMO in the definition of BPMO. Section 4 provides a comparison between the two approaches taken for business process representation in SemBiz and SUPER European project[1]. Section 5 concludes the document.

2 BPMO

The BPMO provides a framework consisting of all the elements needed for defining the business processes. Following the infrastructure proposed by WSMO, the BPMO has the following main elements:

BPMO Ontology The BPMO ontology is a WSMO ontology meant to define the domain of the processes that are to be modeled; as such, it defines the infrastructure for providing the underlying information needed for further modeling the processes.

BPMO Business Process The BPMO Business Process defines the main elements of business processes; it strongly relies on the WSMO Web Service definition, and provides additional support for business process modeling.

BPMO Business Goal The BPMO Business Goal defines the infrastructure for expressing the request for a business process, more exactly the request for a business process functionality.

BPMO Mediators The BPMO Mediators define the types of mediators needed for business process mediation.

Some of these elements were previously described in [10] and [9]. As this document is meant to provide a complete and unified description of BPMO, some definitions will be repeated for consistency reasons.

[1] http://www.ip-super.org/

2.1 BPMO Ontology

The BPMO Ontology is a WSMO Ontology (i.e. it respects the structure and syntax as prescribed by WSMO). Furthermore, the BPMO Ontology is expressed using the WSML representation language.

For defining any business process the domain expert needs to define first all the entities involved in this business process: the organizations, the personnel, the software and the systems used, the business data and so on. The BPMO Ontology provides support for defining all these entities. The concepts that are part of the ontology are listed below:

BusinessRole: defines the entities involved in a business process (interacting with a business process or executing it); this concept has only one attribute, for denoting the name of the BusinessRole;

Actor: sub-concept of BusinessRole, a business organization or a person; the attributes defined for this concept are referring to the software or the physical assets of the Actor;

Organization: sub-concept of Actor, defined for modeling organization specific aspects; instances of this concept must contain information about the location of the organization and at least one employee (the contact person), but they can also specify the web site address, the umbrella organization as well as other employees;

Personnel: sub-concept of Actor, defined for modeling persons involved in a business process; an instance of this concept must specify the organization for which that person works, and it may also specify relevant information such as email address, homepage or title;

Asset: sub concept of BusinessRole, a software system, or a physical asset;

SoftwareSystem: sub-concept of Asset, able to provide a certain functionality; any software system belongs to at least one Actor, which must be specified;

PhysicalAsset: sub-concept of Asset, defining hardware equipments or other belongings (for example credit card); every PhysicalAsset has exactly one location, and belongs to at least one Actor;

Event: the events are used to control and influence business processes;

EventType: used for defining what types of Event are allowed;

Empty: sub-concept of Event, used for marking empty start and end events

StartEvent: sub-concept of Event, denoting the start of a process;

EndEvent: sub-concept of Event, denoting the end of a process;

IntermediateEvent: sub-concept of Event, denoting a particular event occurring during a process;

Message: sub-concept of Event, consisting of a message that may carry business data;

StartMessage: sub-concept of StartEvent and Message, denoting a message that triggers a process;

EndMessage: sub-concept of EndEvent and Message, denoting the end of a process;

IntermediateMessage: sub-concept of IntermediateEvent and Message, a message event occurring during a process execution;

Timer: sub-concept of Event used for measuring time;

StartTimer: sub-concept of StartEvent and Timer, denoting a timer event at the beginning of a process;

EndTimer: sub-concept of EndEvent and Timer, a timer event at the end of a process;

IntermediateTimer: sub-concept of IntermediateEvent and Timer, a timer event occurring during a process execution;

TimePoint: sub-concept of Timer, denoting a particular moment in time; instances of this concept must specify exactly one time value;

TimeDuration: sub-concept of Timer, denoting a particular duration; instances of this concept must specify exactly one interval of time;

Exception: sub-concept of Event, denoting an exception occurring during the execution of a process;

Cancel: sub-concept of Event, which is used for the cancelation of a process' execution;

Log: defined for keeping a record of the process' execution;

BusinessRule: used for defining the rules for a business process, as WSML axioms;

DataElement: used for technically describing the business data;

BusinessData: used for describing the data needed by a business process; instances of this concept must specify the role they belong to and the corresponding data element; in addition, they can specify the name of the business data (if any) and the creation time.

Furthermore, three instances are defined in order to restrict the allowed EventTypes:

START: used for marking events that trigger a process;

INTERMEDIATE: used for marking events that occur during the execution of a process;

END: used for marking events that occur after the execution of a process.

For setting the allowed EventType for the sub-concepts of Event, the following axioms are defined:

axiom allowed_event_types: sets the allowed EventTypes to START, INTERMEDIATE and END;

axiom allowed_emptyEvent_types: sets the allowed EventTypes to START and END;

axiom allowed_startEvent_types: sets the allowed EventTypes to START;

axiom allowed_endEvent_types: sets the allowed EventTypes to END;

axiom allowed_intermediateEvent_types: sets the allowed EventTypes to INTERMEDIATE.

The graphical representation of the concepts and instances defined in the BPMO Ontology using the Web Service Modeling Toolkit[2] is depicted in Figure 1.

[2] http://sourceforge.net/projects/wsmt

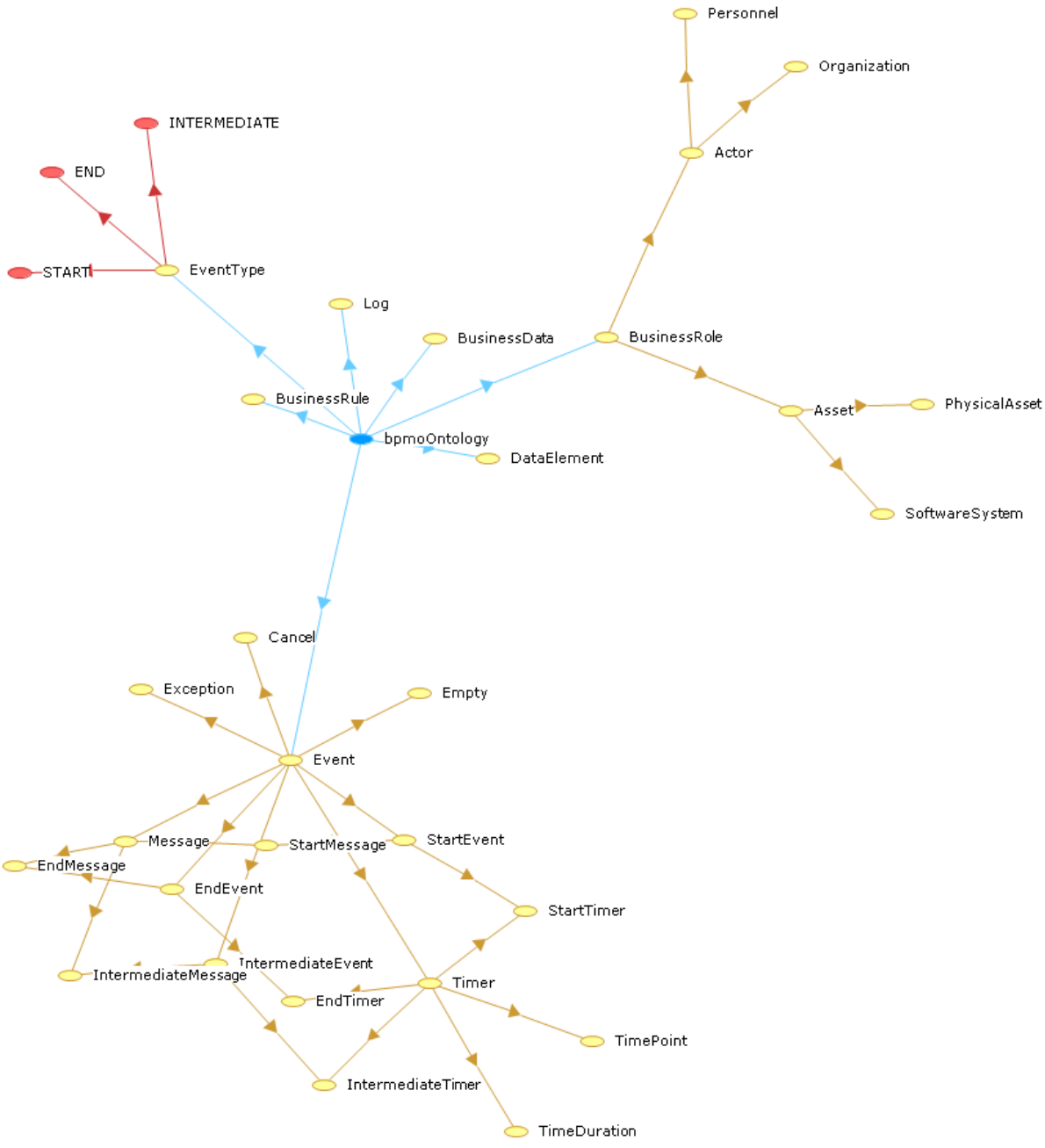

Figure 1: BPMO Ontology

2.2 BPMO Business Processes

The starting point in defining BPMO business processes was the WSMO Web Service definition. Similar to the Semantic Web Services, a business process can import external ontologies and use mediators to overcome heterogeneity problems, it may have a name (which is not mandatory, but a business process can not have more then one name) and it provides a certain functionality (capability).

The main difference between a business process (as defined in SemBiz) and a WSMO Web service is that while the Web Services define an interface consisting of *choreography* and *orchestration* [8], for business processes the focus is on how a service is executed, what roles it interacts with, if there is a service providing that particular functionality and what sub-processes are part of a certain business process.

BPMO Business Process defines the following concepts:

BusinessProcess: main concept of the ontology, defined in terms of the imported ontologies, the mediators used, a name (optional), its capability and its execution;

Capability: defines what a process can do in terms of preconditions (the information space of the process before its execution) and postconditions (the information space of the process after its execution);

Execution: defines the entities involved in the execution of a business process, in terms of the business process' executor, the business roles it interacts with and the IRI of the Web Service able to provide the same functionality as the business process (if any);

CompensationProcess: sub-concept of BusinessProcess, a compensation process is business logic that is called in case of a failure and that reverses the effects of former service calls; this special type of process should not specify a capability, as it is not meant to be discovered by the discovery engine but only linked with the process it compensates for by axioms and relations;

CompositeProcess: sub-concept of BusinessProcess defined as a composition of several business processes; the sub-processes part of a CompositeProcess have to be specified;

AtomicProcess: sub-concept of BusinessProcess that cannot be consider a CompositeProcess;

ParallelProcess: sub-concept of CompositeProcess, with all the composed processes executing in parallel;

SequenceProcess: sub-concept of CompositeProcess, with the composed processes executing in sequence; the sequence is set by using two attributes, one denoting the first process to be executed, the second attribute denoting the sequence of processes (expressed as another SequenceProcess) following the first one;

SelectProcess: sub-concept of CompositeProcess, imposing a selection rule (a BPMO Ontology BusinessRule) for determining which of the composed processes is (are) executed; several types of selections are allowed, as defined by the SelectType concept;

SelectType: defines the allowed select types, by having a predefined set of instances (XOR, OR and CASE);

LoopProcess: sub-concept of CompositeProcess, imposing a loop rule (a BPMO Ontology BusinessRule); several types of loops are allowed, as defined by the LoopType concept;

LoopType: defines the allowed loop types, by having a predefined set of instances (FOR, WHILE_DO and DO_UNTIL).

In addition to the concepts previously listed, the BPMO defines a number of relations, axioms and instances:

relation hasCompensation: this relation links a business process with a compensation process;

relation hasFaultHandler: in addition to linking a business process with a compensation process, this relation also specifies the exception (as BPMO Ontology Exception) that occurred during the execution of the business process

axiom directInstance: defines what means to be a direct instance of a concept; (i.e. created strictly based on the definition of the concept, and not based on the definition of a subconcept of that concept);

axiom setAbstractConceptsConstraint: constraints some concepts to be abstract concepts (i. e., without any direct instances); the restriction is only imposed on the CompositeProcess concept, any instance of the CompositeProcess being created as an instance of its subconcepts;

instances OR, XOR and CASE: define the allowed types of selection;

axiom allowed_select_types: sets the allowed select types to the instances defined above;

instances FOR, WHILE_DO and DO_UNTIL: define the allowed types of loop;

axiom: allowed_loop_types: sets the allowed loop types to the instances defined above.

Due to the on-going work on BPMO business processes translations to the Viewbased modeling framework (VbMF) [3] needed for the execution of business processes, several changes in the representation of the composite processes may be still needed. If this will happen, the proposed changes will be evaluated and properly documented in the forthcoming deliverables.

The graphical representation of the concepts and instances defined in the BPMO Business Process using the Web Service Modeling Toolkit is depicted in Figure 2.

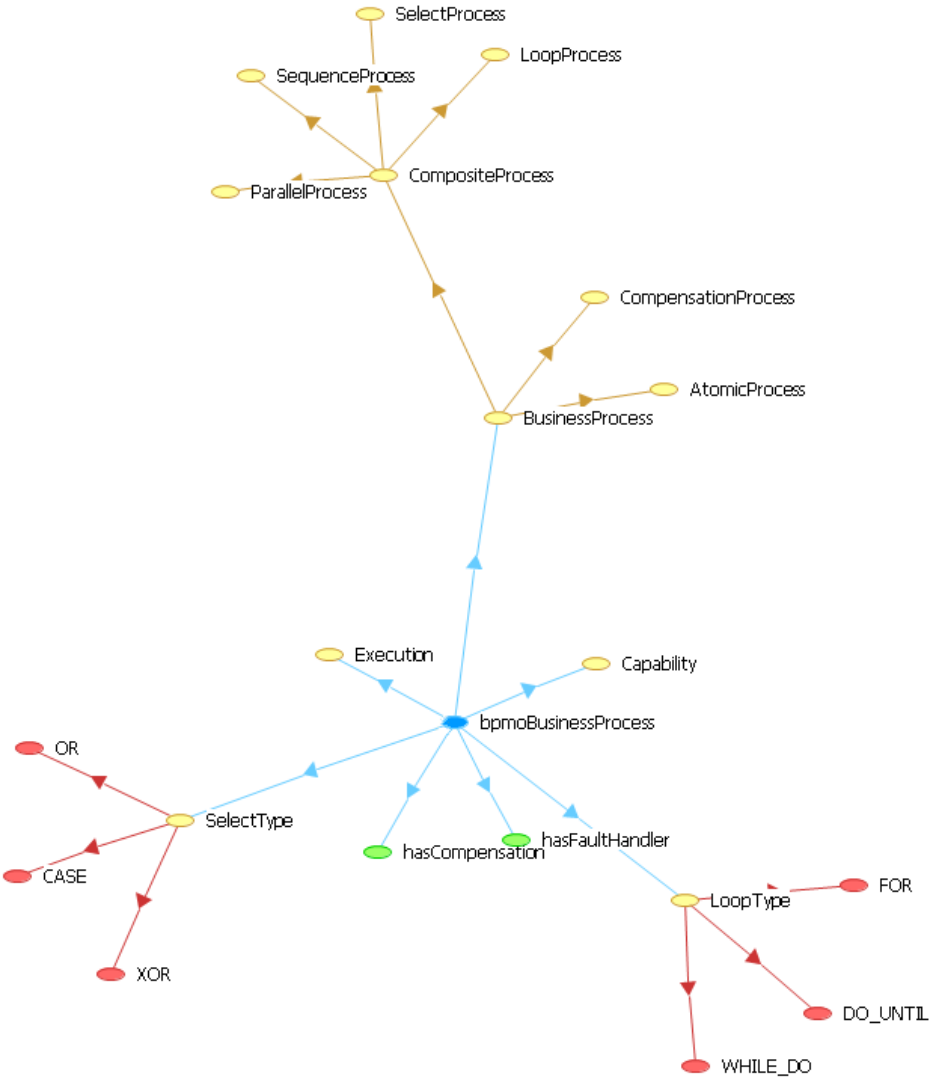

Figure 2: BPMO Business Process

2.3 BPMO Business Goals

A BPMO business goal defines what a requestor wants to achieve after the execution of a business process. As such, its definition has to be similar with a business process definition. The main difference is that the requestor of a process is not interested in how a process is executed, but only in the provided capability. Another difference is that the type of the process (composite or atomic) able to fulfil a goal is again not relevant.

The BPMO Business Goal consists only of the following element:

BusinessGoal: main concept of the ontology, defined in terms of the imported ontologies, the mediators used, a name (optional), and the requested capability (with range of type Capability as defined in the BPMO Business Process).

2.4 BPMO Mediators

BPMO does not define specific mediators, but reuses the definitions of the WSMO mediators considered relevant for Business processes.

A WSMO mediator, as defined in [8] consists of the following elements:

imported ontologies: the ontologies used for the definition of the mediators;

non functional properties: define the non-functional characteristics of the mediator;

source and target: for defining the elements the mediator mediates between;

mediation service: the actual mediation service able to mediate between the source and the target.

Furthermore, WSMO defined four types of mediators, as described in [6] for ontology-to-ontology mediation (ooMediator), goal-to-goal mediation (ggMediator), Web service-to-Web service mediation (wwMediator) and Web service-to-goal mediation (wgMediator). The last two types of mediators consider mediation from two perspectives: that of capability and of behavior (as defined by the interface of the Web services and goals). As BPMO does not include these aspects, we are only interested in the first two definition: ooMediators and ggMediators. The complete definitions and examples on how to use these mediators can be found at [6].

3 BPMO and WSMO

The Business Process Modeling Ontology relies strongly on the Web Services Modeling Ontology, a well established framework for describing all aspects related with Semantic Web Services. Firstly, BPMO adopts the four main modeling elements: Ontologies, Web Services, Goals and Mediators. Secondly, the definitions provided by WSMO were updated in order to be used for the business processes domain, as the purpose of BPMO is to semantically define business processes, and not Semantic Web Services. The main differences between the two frameworks are given by this separation of scope: business processes versus Web services.

The relation between the BPMO elements and the WSMO elements is a s follows:

Ontology: The BPMO ontology is a WSMO ontology defined specifically for the business process domain. As such, WSMO ontology can be seen as a meta-level for the BPMO ontology, defining the allowed elements in terms of concepts, instances, relations and axioms. Taking these definitions as a starting point, the BPMO defines specific elements needed for the domain ontology of a business process.

Business Processes: The definition of the business processes relies on WSMO Web service definition. The main difference between the Web services and processes definition is that while the first is concerned with the interface of the service (how can a service be invoked and how it can communicate with other services), for the latter an important aspect is how the process is executed and what roles (humans, organizations, software or hardware) are involved in the execution. Another important difference lies in the classification of processes as either atomic or composed services, which is not relevant for the Semantic Web Services.

Business Goal: The scope of the goal as defined in BPMO is the same as the scope of the WSMO goal: to express what a requestor wants to achieve after the execution of a business process or service. The difference is given by the fact that a requestor of a business process is not interested in how the process can be invoked or in how it can execute (as opposed to the requestor of a Web service).

Mediators: The definition of WSMO mediators was adopted by BPMO. However, the mediators specifically designed for dealing with Web services are not relevant in this case. Instead, as the business processes are defined without using a specific behavioral formalism, the ooMediators can be used for defining process to process mediation in terms of process hierarchies and the ggMediators can be used for mediating between capabilities.

4 SUPER Business Process Modeling and Execution

The SemBiz M12 Review [7] recommended that

> The BPMO ontology developed in Sembiz should be integrated with the Process Modeling Ontology developed in SUPER and used as a basis for the System.

We now provide a brief summary of the overall similarities between the two projects, and justify our inability to meet the recommendation.

4.1 Comparison between SemBiz and SUPER

Both SUPER and SemBiz are concerned with the application of logical semantics to the modeling of business processes and the semi-automatic transformation of such models to an executable representation. While sharing many overall similarities, the important differences for the recommended use of SUPER BPMO in SemBiz include:

1. SUPER aims to construct a purely ontological translation between a business process model and its executable representation. SemBiz follows a hybridized approach where

semantic models are first translated into object oriented constructs available from the Eclipse Modeling Framework, and then further transformed by purely syntactical methods to standardized BPEL.

2. The BPEL produced by SemBiz is intended to be able, with minor modifications to the VbMF pipeline, to be deployable to any generic BPEL container. In contrast, SUPER uses a semantically enhanced runtime that utilizes the abstraction of the SEE (Semantic Execution Environment) to allow runtime binding of Web services and the collection of semantically annotated execution logs.

4.2 Remarks on the use of SUPER BPMO in SemBiz

At this time, SemBiz M18[3], we are unable to carry out the the recommendation on the reuse of SUPER BPMO, as the SUPER BPMO – undergoing weekly if not daily revision – is currently not stable enough to utilize. Additionally, BPMO is the site of considerable contestation among members of the SUPER consortium, with widely divergent views being espoused as to what constructs are necessary for the proposed purely ontological transformation to executable BPEL.

This contestation does not necessarily preclude the use of SUPER BPMO, although it begs the question as to which version one would use. The more significant problem is that SUPER BPMO has not been validated by the existence of a transformation pipeline. The work on such a transformation is not due until SUPER M36, and is currently not proposed to start before SUPER M30. Simply put, there is no way to determine if the SUPER BPMO as currently envisioned will work for SUPER, let alone be applicable to the syntactical transformation pipeline within the SemBiz VbMF.

The SUPER BPMO is the ontologization of a common superset of graphical notations including BPMN, EPC, and possibly other graph based notations. SemBiz BPMO is an ontologization of the concrete graphical notation as defined by the code that runs in the SemBiz WSMT based modeler (based in turn on the abstractions afforded by the Eclipse GEF). While SUPER should theoretically be able to abstract an additional graphical notation, in practice, each new notation that has been formalized has required refactoring and modification of BPMO.

4.3 Other Possibilities

While it is not currently possible to utilize the SUPER BPMO within SemBiz, there are other possibilities for utilizing other components of SUPER within SemBiz:

1. Considerable effort within SUPER has been expended in the ontologization of the eTOM/NGOSS/SID abstractions made available by the Tele-Management Forum. This effort within SUPER is known collectively as YATOSP (Yet Another Telecoms Ontology, Service and Process Framework)[1]. YATOSP currently consists of three separate ontologies: an ontology describes in details the vocabulary of a domain, inside the telecommunication domain (SID - Shared Information and Data Model) which describes

[3]SemBiz is a twenty-four month project, SUPER is a thirty-six month project, so SemBiz M18 is roughly SUPER M23

the concrete entities, the operation map (eTOM - enhanced Telecommunication Operation Map) which in its turn is related with the business goal inside a telecommunication system and the Application Map (TAM - Telecommunication Applications Map). The reuse of these ontologies within SemBiz would provide a rich set of ontological inputs for the Query and Discovery engine when applied to the eTel Use Case scenario.

2. Feedback from SemBiz to SUPER is actually a much more likely scenario. Even though the funding and resources of SUPER dwarfs that of SemBiz, the geographical proximity of the consortium members along with the more focused and less ambitious research agenda has actually produced more concrete results at this point than SUPER, such as a more-or-less complete toolchain for the execution of a prototype. Such equivalent work for SUPER will probably not occur until SUPER M26 (SemBiz M22).

Despite the consortium's current inability to meet the specific request to utilize the SUPER BPMO in SemBiz, we understand the spirit in which the request was made to utilize the results of the much large European-wide Integrated Project within SemBiz. Now that the first iteration of the SemBiz toolchain has been implemented in the eTel prototype, we will ruthlessly look to plunder and reuse what stably exists in SUPER for the second iteration of the implementation of the SemBiz toolchain.

5 Conclusions

This document presents the BPMO and its four main elements: ontologies, business processes, goals and mediators. The previous versions of BPMO [10][9] underlined the state of the art and the decisional process for defining all the BPMO elements, and this parts were omitted in this document. Also the evaluation conducted as part of the deliverable [2] was taken into consideration when finalizing the BPMO.

Furthermore, this document is accompanied by the three wsml files containing: the BPMO ontology (used for modeling the domain of the business process), the BPMO business process model (providing the infrastructure needed for modeling specific processes) and the BPMO goal definition (offering support for the declaration of a business request). As the mediators definition was adopted from the WSMO specifications, the corresponding wsml document is not provided here; for more details about the mediators we refer the reader to [6].

The relationship between WSMO specifications and BPMO is also presented in this document, with emphasize on how BPMO was influenced by WSMO modeling and design principles and definitions, in the same time offering additional support for business process modeling. The relationship between the SemBiz approach and the one taken in SUPER European project is also presented.

References

[1] S. Arroyo, L. Cicurel, M. Evenson, D. de Francisco, H. Muoz, P. Rubach, E. Szczekocka, and P. Todorova. SUPER D8.2 YATOSP Framework. SUPER Deliverable, August 2007.

[2] E. Cimpian, S. Dustdar, J. Hoffmann, T. Holmes, M. Kux, A. Sirbu, , and U. Zdun. D5.1 Exploratory Case Study. SemBiz Deliverable, November 2007.

[3] S. Dustdar, J. Hoffmann, T. Holmes, A. Sirbu, H. Tran, and U. Zdun. D1.2 Semantic Querying, Discovery, and Composition Framework. SemBiz Deliverable, August 2007.

[4] N. Guarino and C. Welty. A formal ontology of properties. In *Proceedings of EKAW-2000: The 12th International Conference on Knowledge Engineering and Knowledge Management*, Berlin, 2000.

[5] N. Guarino and C. Welty. An overview of ontoclean. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 151–159. Springer, New York, 2004.

[6] A. Mocan, E. Cimpian, MichaelStollberg, F. Scharffe, and J. Scicluna. WSMO Mediators. WSMO Working Draft D29v0.2, available at: `http://www.wsmo.org/TR/d29/v0.2/`, December 2005.

[7] Reviewers. Review report semantics business process management for flexible dynamic value chains (sembiz). https://trac.sti2.at/svn/Review/ReviewReport.pdf, 2007.

[8] D. Roman, H. Lausen, and U. Keller. D2v1.3. web service modeling ontology (wsmo). WSMO Deliverable available from http://www.wsmo.org/TR/d2/v1.3/, October 2006.

[9] Z. Yan, E. Cimpian, and T. Holmes. D1.2 Business Process Modeling Ontology (BPMO) version 1. SemBiz Deliverable, September 2007.

[10] Z. Yan, E. Cimpian, M. Mazzara, and M. Zaremba. D1.1 BPMO Requirements Analysis and Design. SemBiz Deliverable, February 2007.