

D1.2. Business Process Modeling Ontology (BPMO) version 1

Emilia Cimpian¹, Zhixian Yan¹, Ta'id Holmes²

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

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



Abstract

Semantical description of business processes is one of the most challenging research issues currently addressed by the semantic community. This is because the development of a conceptual model for semantic business processes requires a joint effort of business experts and academia and it needs to go through several phases of modeling and testing. This deliverable proposes a Business Process Modeling Ontology, which contributes to bridging the gap between the business level perspective and the technical implementation level in Business Process Management by semantic descriptions of business processes.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 2 |
| 2 | BPMO Engineering | 2 |
| 3 | Business Process Modeling Ontology | 3 |
| 3.1 | BPMO Main Elements | 3 |
| 3.2 | BPMO Ontology | 4 |
| 3.2.1 | Business Role | 5 |
| 3.2.2 | Events | 7 |
| 3.2.3 | Log | 8 |
| 3.2.4 | Business Rule | 8 |
| 3.3 | BPMO Process | 9 |
| 3.3.1 | Business Process | 9 |
| 3.3.2 | Capability | 11 |
| 3.3.3 | Execution | 11 |
| 3.3.4 | SelectType and LoopType | 12 |
| 3.4 | Business Goal | 13 |
| 3.5 | Mediators | 13 |
| 3.6 | Integration with EMF | 14 |
| 3.6.1 | View-based modeling framework | 14 |
| 4 | Conclusion | 17 |
| A | Product Ordering Use Case | 18 |

1 Introduction

One of the main objectives of the SemBiz project is to develop a semantic description framework for business processes that allows business process management by business experts on a higher level of abstraction, as well as support for automated handling and execution of business processes on the technical level. For this, the consortium is developing an ontology, namely the Business Process Modeling Ontology (BPMO), further presented in section 3

For developing the BPMO we followed the main steps for ontology engineering, as further presented in Section 2. Section 3 presents the Business Process Modeling Ontology, with its main elements: ontologies, business processes, mediators and business goals.

This deliverable also presents the business process modeling ontology corresponding to the product ordering use case, in Annex A. A complete description of the use case is provided in deliverable 4.1 Use Case Definition and Functional Requirements Analysis [9].

2 BPMO Engineering

Several methodologies for ontology engineering have been elaborated during the last few decades. In [5] the author conducted a study based on more than twenty such methodologies, concluding that the majority of them propose six steps for ontology creation and maintenance, as follows:

1. Domain/Requirements analysis;
2. Conceptualization;
3. Implementation;
4. Evaluation;
5. Population;
6. Evolution and maintenance.

The first phase, *domain analysis*, consists of the detailed analysis of the domain that is going to be modeled, having as result a clear understanding of those aspects of the domain that are going to be represented in the ontology, and a set of requirements. This step also includes knowledge acquisition in terms of re-usage of existing ontological sources or performing ontology learning operations. For BPMO this phase was performed in the first six months of the project, having as result deliverable 1.1 BPMO Requirements Analysis and Design [14].

The second step, *conceptualization*, consists of the definition of the concepts, relations and instances that are going to be used in representing the domain; the result is the corresponding conceptual model. The conceptualization phase for BPMO also started in the first semester of the project, and continued in the second semester. The results are presented in this deliverable, in the following chapter.

The *implementation* phase consists of representing the conceptual model in an ontology representation language with the adequate expressivity. The language chosen for BPMO representation is the Web Service Modeling Language (WSML¹), and the rationale behind this choice was presented in [14].

The fourth phase consists of the *evaluation* of this initial ontology against the previously identified requirements. The evaluation may trigger modifications of the conceptual model, and as a consequence of the implementation as well. The evaluation of this first version of BPMO will be performed in Work Package 5 (Validation and Evaluation), where an exploratory use case will be developed. Based on the result of this evaluation, BPMO will be extended and possibly modified.

The *population* phase deals with the alignment of concrete application data to the implemented ontology. This phase will consist of a complete modeling of the use cases using BPMO, as part of Work Packages 1 and 4. The use case modeling is a joint effort between the industrial partners, who are familiar with the technical aspects of the use cases, and the academic partners, who provide their experience in semantics and business process modeling technologies.

Further modifications or even complex re-engineering tasks are performed in the final phase, *evolution and maintenance*. In the SemBiz project, this phase coincides with the developments of the second version of BPMO.

3 Business Process Modeling Ontology

The initial proposal for the BPMO was made during the first six months of the project, and the results were presented in [14]. This section elaborates on the ontology definition, with several updates and improvements. It first presents the main BPMO elements (Section 3.1), and continues with the elaboration elaborating on these main elements, with one sub-section dedicated to each of them.

3.1 BPMO Main Elements

In defining BPMO we had to consider the main aspects that need to be represented in order to allow the automatic discovery, composition and execution of processes. The main element that needs to be represented is, of course, the *business process*. In defining a process one may need to use an underlying *ontology*, so the ontologies is the second type of BPMO elements that need to be modeled. The request (*business goal*) for a business process represents the third type of elements, and the goals definition also needs the support of the BPMO ontologies. As there may be heterogeneity problems between different elements, semantic mediators may be needed - the fourth type of modeling elements. All of these elements can be characterized by a list of *non-functional properties*, as the one described in [14].

Figure 1 presents a graphical representation of the BPMO main elements. Each of them will be further detailed in the following sections.

¹www.wsmo.org/wsml

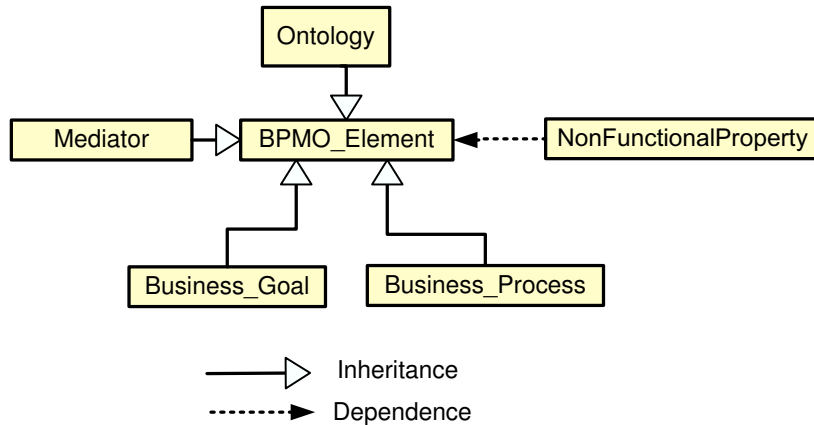


Figure 1: Upper BPMO Elements

3.2 BPMO Ontology

The BPMO ontology is a WSMO² ontology (Web Service Modeling Ontology), as defined in [12]. The definition, taken from [12] is depicted in Listing 1.

Listing 1: WSMO ontology Definition

```

Class ontology sub-Class bpmoElement
importsOntology ofType ontology
usesMediator ofType ooMediator
hasConcept ofType concept
hasRelation ofType relation
hasFunction ofType function
hasInstance ofType instance
hasRelationInstance ofType relationInstance
hasAxiom ofType axiom
  
```

The *wsmoElement* inherited by the *ontology* encapsulates the most generic definition which applies to all WSMO elements. Any WSMO ontology consists of *concepts*, *relations*, *functions*, *instances*, *relationInstances* and *axioms*. Additionally, an ontology can import other ontologies, and can use *mediators* in case there are heterogeneity mismatches. All of these are defined in detail in [12].

As BPMO ontology is meant to offer support for process modeling, a number of concepts that have to be part of this ontology were identified, namely *business role*, *business rule*, *event* and *log*. Sub-concepts hierarchies were created for the *business role* and *event*, as shown in Figure 2³.

²www.wsmo.org

³the diagram was generated using the Web Service Modeling toolkit [10]

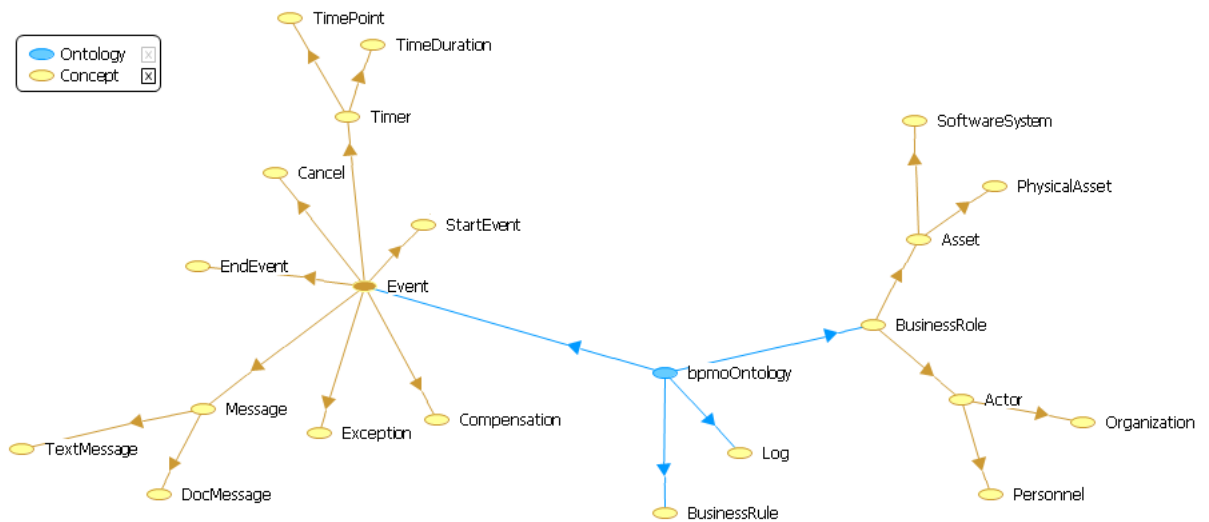


Figure 2: BPMO Ontology

3.2.1 Business Role

The business role is one of the most important concept of the BPMO ontology, and it defines the entity (human or organization) that interacts with or performs a certain process. The link between the process and the business role is realized in the process' definition. There are several sub-concepts defined for the business role, as shown in Listing 2

Listing 2: The *BusinessRole* Concept and its Sub-Concepts

```

wsmIVariant _ "http://www.wsmo.org/wsmI/wsmI-syntax/wsmI-flight"
namespace { _ "http://www.sembiz.org/bpmoOntology#",
            dc _ "http://purl.org/dc/elements/1.1#"
}

ontology bpmoOntology
  nonFunctionalProperties
    dc#title hasValue "BPMO Ontology"
    dc#language hasValue "en-US"
    dc#subject hasValue {"Semantic Business Process Modeling",
                        "BPMO Model", "WSMO"}
    dc#format hasValue "text/html"
  endNonFunctionalProperties

concept BusinessRole
  nonFunctionalProperties
    dc#description hasValue "BusinessRole: perform and interact with
                            business processes"
  endNonFunctionalProperties
  hasName ofType (1) _string

concept Actor subConceptOf BusinessRole
  nonFunctionalProperties
    dc#description hasValue "Actor: a special BusinessRole, as business
                            organization, certain person etc."
  endNonFunctionalProperties

```

```

hasSoftware ofType (0 *) SoftwareSystem
hasPhysicalAsset ofType (0 *) PhysicalAsset

concept Organization subConceptOf Actor
nonFunctionalProperties
    dc#description hasValue "Organization: a special BusinessRole, as
        business organization"
endNonFunctionalProperties
hasWebsite ofType _string
hasLocation ofType _string
hasContactPerson ofType Personnel
hasMember inverseOf(inOrganizations) ofType (0 *) Personnel
hasSuperGroup ofType (0 *) Organization

concept Personnel subConceptOf Actor
nonFunctionalProperties
    dc#description hasValue "Personnel: a special BusinessRole,
        people involved in the processes"
endNonFunctionalProperties
hasTitle ofType _string
hasHomepage ofType _string
hasEmail ofType _string
inOrganization inverseOf(hasMember) ofType (0 *) Organization

concept Asset subConceptOf BusinessRole
nonFunctionalProperties
    dc#description hasValue "Asset: SoftwareSystem, PhysicalAsset"
endNonFunctionalProperties

concept SoftwareSystem subConceptOf Asset
nonFunctionalProperties
    dc#description hasValue "SoftwareSystem: a special BusinessRole,
        defining systems like billing, crm systems"
endNonFunctionalProperties
belongTo inverseOf(hasSoftware) ofType Actor

concept PhysicalAsset subConceptOf Asset
nonFunctionalProperties
    dc#description hasValue "PhysicalAsset: a special BusinessRole, defining
        hardware equipments or other personal
        privacy belongings like credit card, account"
endNonFunctionalProperties
belongTo inverseOf(hasPhysicalAsset) ofType Actor
inLocation ofType _string

```

WSML recommends the use of XML Schema datatypes as defined in [4] for the representation of concrete values, such as strings and integers [6]. Values of these most primitive datatypes can be used to construct values of more complex datatypes, e.g. date and time.

As presented in Listing 2, an *BusinessRole* can be an *Actor - Personnel* or *Organization* - or an *Asset - Physical Asset* or *Software System*. The distinction between all these elements is given by who can play a business role, and what kind of role that is: execute a business process, contribute to its execution and so on. Further more, every element has some attributes that help in defining it as an entity, such as contact information for any actor or the corresponding web site for an organization. Additionally, some relations can be defined between these elements, for example *hasMember* and *inOrganization* for defining what instances of personnel belong to a certain organization.

Several new initiatives have defined generic human roles in relation with a process' tasks [3], [2]. These roles will be taken into consideration for the next version of BPMP, with respect to

the challenges raised by the use cases. For now, the set of concepts currently defined in BPMO is expressive enough for the existing use cases.

3.2.2 Events

In terms of process modeling, events are entities that describe the situation before and after a function is executed [1]. Based on the use case specification and on the state of the art analysis, we defined the concept *event* as well as a set of sub-concepts, as presented in Listing 3.

Listing 3: The *Event* Concept and its Sub-Concepts

```
concept Event
  nonFunctionalProperties
    dc#description hasValue "concept describing an event"
  endNonFunctionalProperties
  hasName ofType _string
  hasTextContent ofType _string
  creationTime ofType _time
  belongsTo ofType BusinessRole

concept Message subConceptOf Event
  nonFunctionalProperties
    dc#description hasValue "particular event, consisting of a message"
  endNonFunctionalProperties

concept TextMessage subConceptOf Message
  nonFunctionalProperties
    dc#description hasValue "text message (like a memo)"
  endNonFunctionalProperties

concept DocMessage subConceptOf Message
  nonFunctionalProperties
    dc#description hasValue "message with attachements"
  endNonFunctionalProperties

concept Timer subConceptOf Event
  nonFunctionalProperties
    dc#description hasValue "timer concept"
  endNonFunctionalProperties

concept TimePoint subConceptOf Timer
  nonFunctionalProperties
    dc#description hasValue "time point"
  endNonFunctionalProperties
  timePointValue ofType _time

concept TimeDuration subConceptOf Timer
  nonFunctionalProperties
    dc#description hasValue "duration"
  endNonFunctionalProperties
  durationInHours ofType _integer
  durationInDays ofType _integer
  durationInMinutes ofType _integer

concept StartEvent subConceptOf Event
  nonFunctionalProperties
    dc#description hasValue "particular event denoting the start
      of an action/process"
  endNonFunctionalProperties

concept EndEvent subConceptOf Event
  nonFunctionalProperties
```

```

        dc:description hasValue "particular event denoting the end
                                of an action/process"
    endNonFunctionalProperties

concept Exception subConceptOf Event
    nonFunctionalProperties
        dc:description hasValue "exception in a process execution"
    endNonFunctionalProperties

concept Cancel subConceptOf Event
    nonFunctionalProperties
        dc:description hasValue "cancelation of an execution"
    endNonFunctionalProperties

concept Compensation subConceptOf Event
    nonFunctionalProperties
        dc:description hasValue "compensation"
    endNonFunctionalProperties

```

As shown in Listing 3, there are five special types of events, namely *StartEvent*, *EndEvent*, *Exception*, *Cancel* and *Compensation*. Instances of the *StartEvent* concept can be used for marking the beginning of a process, while the instances of the *EndEvent* concept mark the finalization of a process. *Cancel* can be used for stopping a process (for example due to an unexpected course of events), while instances of *Exception* and *Compensation* can be used for marking and recovering after an error.

3.2.3 Log

Instances of this concept are created during the execution of a process, with the sole purpose of maintaining a record. Any errors that may appear during the execution are also recorded here. So far, we did not envision any sub-concepts needed for this concept.

Listing 4: The *Log* Concept Definition

```

concept Log
    nonFunctionalProperties
        dc:description hasValue "instances of this concept are automatically
                                generated during a process' execution"
    endNonFunctionalProperties
    hasTextContent ofType _string

```

3.2.4 Business Rule

Instances of this concept define the rules that govern the execution of the business process. This concept is still under-specified, and it will be further elaborated in the sub-sequent version of BPMO.

Listing 5: The *BusinessRule* Concept Definition

```

concept BusinessRule
    nonFunctionalProperties
        dc:description hasValue "concept describing BusinessRule"

```

```

endNonFunctionalProperties
hasName ofType _string
hasTextMemo ofType _string
hasFormalMemo ofType _string

```

3.3 BPMO Process

In modeling a business process we have to take into consideration several aspects, namely the functionality (*capability*) of a process, its *executions*, the type of process (*atomic* or *composite*) and the allowed types of composing process (*select*, *loop* or *parallel*). Hierarchies of concepts and instances were created in order to offer full support for process modeling, as shown in Figure 3.

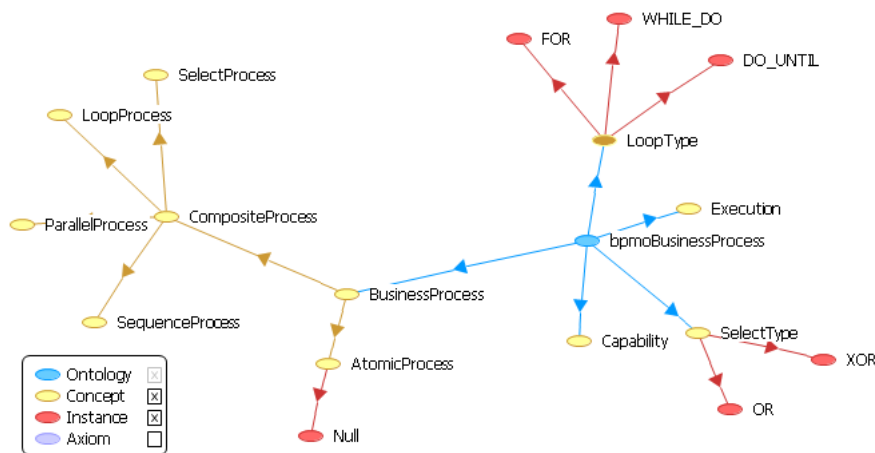


Figure 3: BPMO Process Ontology

3.3.1 Business Process

Taking into consideration the state of the art survey in process representation conducted in [14], we classified the processes in two categories: atomic processes (consisting of a single activity, or a single execution step) and composite processes, consisting of several atomic and composite processes. The composite processes can be of multiple types, like parallel, sequential, selection of processes and loops. Furthermore, the processes are characterized by their functionality (also called capability) and execution (further defined in the following sections).

AtomicProcess concept For defining an atomic process the only elements of interest are the capability and the execution.

CompositeProcess concept The definition of a composite process, needs to contain, in addition to the capability and execution definition, a list of its sub-processes.

ParallelProcess concept A special type of composite process, where all the sub-processes are considered to be executed in parallel.

SequenceProcess concept For defining a sequence of processes one needs to specify the order of the sub-processes. For marking the end of the sequence, a special instance of the atomic process, the *Null* instance, was created.

SelectProcess concept In this case only some of the sub-concepts are executed, based on a selection rule. The allowed types of selections are described in Section 3.3.4.

LoopProcess concept A sequence of processes is executed multiple times, depending on a given condition. The allowed types of loops are described in Section 3.3.4.

A complete description of the processes is provided in Listing 6.

Listing 6: The *BusinessProcess* Concept and its Sub-Concepts

```

wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://www.sembiz.org/bpmoBUsinessProcess#" ,
            dc _"http://purl.org/dc/elements/1.1#" ,
            bpmo _"http://www.sembiz.org/bpmoOntology#" }

ontology bpmoBusinessProcess

    importsOntology
        _"http://www.sembiz.org/bpmoOntology#bpmoOntology"

concept BusinessProcess
    nonFunctionalProperties
        dc#description hasValue "concept describing a Business Process "
    endNonFunctionalProperties
    name ofType _string
    withExecution ofType Execution
    withCapability ofType Capability

concept AtomicProcess subConceptOf BusinessProcess
    nonFunctionalProperties
        dc#description hasValue "concept describing AtomicProcess"
    endNonFunctionalProperties

instance Null memberOf AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "instance used for marking the end of a
                                sequence of processes"
    endNonFunctionalProperties

concept CompositeProcess subConceptOf BusinessProcess
    nonFunctionalProperties
        dc#description hasValue "concept describing CompositeProcess"
    endNonFunctionalProperties
    subprocess ofType (1 *) BusinessProcess

concept ParallelProcess subConceptOf CompositeProcess
    nonFunctionalProperties
        dc#description hasValue "the sub-processes of a parallel process
                                are all executed in parallel"
    endNonFunctionalProperties

concept SequenceProcess subConceptOf CompositeProcess
    nonFunctionalProperties
        dc#description hasValue "the sub-processes of a sequence process are
                                all executed in sequence"
    endNonFunctionalProperties
    tailProcess ofType {SequenceProcess, AtomicProcess}

```

```

    headProcess ofType BusinessProcess

concept SelectProcess subConceptOf CompositeProcess
  nonFunctionalProperties
    dc:description hasValue "only some of the sub-processes of a select
                                process are executed"
  endNonFunctionalProperties
  hasRule ofType bpmo#BusinessRule
  hasSelectType ofType { _string , SelectType}

concept LoopProcess subConceptOf CompositeProcess
  nonFunctionalProperties
    dc:description hasValue "repeating a sub-process until some condition
                                is met"
  endNonFunctionalProperties
  hasRule ofType bpmo#BusinessRule
  hasLoopType ofType { _string , LoopType}

```

The fact that the *businessProcess* concept, its sub-concepts and instances are modeled in a different ontology than the one presented in Section 3.2 is simply a modeling choice. The concepts defined in the BPMO ontology are going to be used also in the definition of the process requests. In case concepts from one ontology are used in another ontology, the first one needs to be imported and its concepts are referred to using an identifier (like *bpmo* from *bpmo#BusinessRule*).

3.3.2 Capability

The *capability* of a process expresses what the service can do, in terms of inputs, outputs, preconditions and postconditions. The inputs and outputs are events (as defined in Section 3.2.2) that trigger the execution of the process, or are generated as a result of the execution. The pre and post conditions impose additional restrictions on these inputs and outputs, and they are expressed as WSMO axioms.

Listing 7: The *Capability* Concept Definition

```

concept Capability
  nonFunctionalProperties
    dc:description hasValue "describes what a process does, in terms of
                                inputs, outputs, preconditions
                                and postconditions"
  endNonFunctionalProperties
  hasInput ofType bpmo#Event
  hasOutput ofType bpmo#Event
  // _iri represents the iri of a WSMO axiom
  hasPrecondition ofType _iri
  hasPostcondition ofType _iri

```

3.3.3 Execution

The *execution* concept provides information regarding the actual execution of a process. It identifies the entities involved in the process, the processes that need to be executed before, and provides the grounding to a service that actually executes that process (Listing 8).

Listing 8: The *execution* Concept Definition

```

concept Execution
  nonFunctionalProperties
    dc#description hasValue "contains information relevant for the execution
                                of the process (who is involved in the
                                executions , what processes need to be
                                executed before this one , what Web Service
                                actually executes the process)"
  endNonFunctionalProperties
  hasPreProcess ofType BusinessProcess
  executedBy ofType bpmo#BusinessRole
  interactsWith ofType bpmo#BusinessRole
  // _iri represents the iri of a WSMO Web Service
  hasGrounding ofType _iri

```

3.3.4 SelectType and LoopType

The *SelectType* and *LoopType* concepts are used for defining the allowed constructs for selections and loops. This is done by using the corresponding instances and the two axioms from Listing 9

Listing 9: The *SelectType* and *LoopType* Concepts Definition

```

concept SelectType
  nonFunctionalProperties
    dc#description hasValue "used for describing what types
                                o
                                f selections are allowed"
  endNonFunctionalProperties

concept LoopType
  nonFunctionalProperties
    dc#description hasValue "used for describing what types
                                of loops are allowed"
  endNonFunctionalProperties

instance XOR memberOf SelectType
instance OR memberOf SelectType
instance FOR memberOf LoopType
instance WHILE.DO memberOf LoopType
instance DO.UNTIL memberOf LoopType

axiom allowed_select_types
  definedBy
    !- ?x memberOf SelectType and
    naf ((?x = XOR or ?x = OR)).

axiom allowed_loop_types
  definedBy
    !- ?x memberOf LoopType and
    naf ((?x = FOR or ?x = WHILE.DO or ?x = DO.UNTIL)).

```

When selecting between several processes, the constructs *OR* and *XOR* (exclusive or) are allowed - based on the condition, only some processes can be executed (please note that this processes can be composite ones). Regarding the loops, BPMO allows the constructs *FOR*,

WHILE_DO and *DO_UNTIL*, which state that the composed processes are executed a number of times, while a condition holds, or until a condition is fulfilled.

3.4 Business Goal

The only thing of importance in modeling the business request is the requested functionality. As a consequence, the business goal can be modeled as in Listing 10, where the requested capability has the same definition as in Section 3.3.2.

Listing 10: The *Business Goal* Concept Definition

```
concept BusinessGoal
  nonFunctionalProperties
    dc:description hasValue "concept describing Business Goal"
  endNonFunctionalProperties
  withCapability ofType bpmoPr#Capability
```

Using the same definition of capability for both the business process definition and the business request facilitates the discovery process.

3.5 Mediators

Defining mediators for business processes is out of the scope of the semBiz project. However, the possibility that a business goal and the matching business process use different underlying ontologies, or that different processes that need to be composed are expressed in different terms, exists. For this, we consider necessary to introduce an additional element, namely the *mediators*, which are meant to overcome the heterogeneity problems. For modeling the mediators, we adopt the WSMO definition of ontology to ontology mediators, *ooMediators* (Listing 11), with the restriction that both the source and the target are ontologies.

Listing 11: The *ooMediator* Definition

```
Class ooMediator sub-Class mediator
  hasNonFunctionalProperties type nonFunctionalProperties
  importsOntology type ontology
  hasSource type {ontology, ooMediator}
  hasTarget type {ontology, Goal, webService, ooMediator}
  hasMediationService type {Goal, webService, wwMediator}
```

Based on this definition, a mediator provides a link (or bridge) between two ontologies, the source and the target ontology. The *ooMediator* provides just an ontological description of the mediator, while the actual mediation task is performed by a mediation service [11].

3.6 Integration with EMF

A supplementary ontology has been defined for BPMO within meta-models of the eclipse modeling framework (EMF) [8]. This is in order to have object oriented representations of business processes on meta-levels from which executable processes can be derived from.

The ontology thus, as expressed by these meta-models, directly emerges from the foundations of service oriented computing unlike the WSML extended one as defined in section 3 and defines basic elements for process design and composition.

A model driven development framework states the linkage of ontologies from the semantic to the object oriented world and performs model-to-model and -code transformation for process composition for final deployment. Model elements, when populated, serve as the source for process composition.

3.6.1 View-based modeling framework

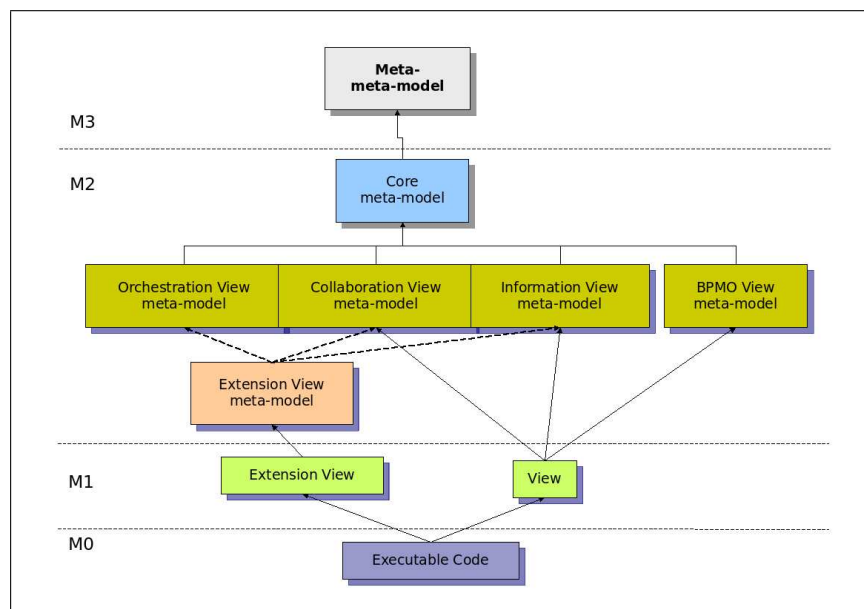


Figure 4: View Based Modeling Framework

Figure 4 gives an overview of a framework that consists of a set of views of distinct concerns [13]. We will present the framework in detail in [7].

Integration of WSML into EMF, holding the completing ontology for process composition, can be realized using a new view from which the other views get populated after model to model transformation so that final process composition can take place by the model driven development framework.

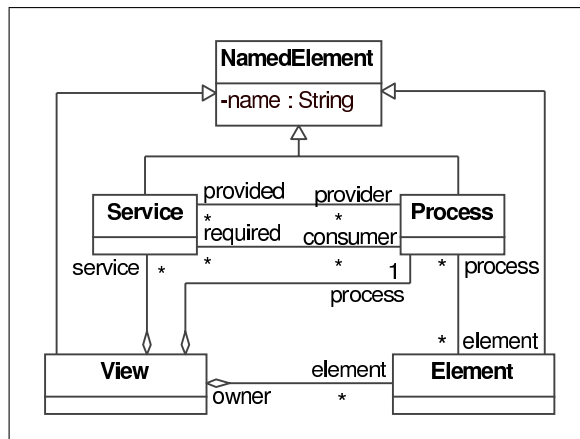


Figure 5: The Core Meta-Model

Core meta-model The core meta-model, from which all views are derived from, is presented in figure 5 and defines extensible elements for process design. The core meta-model is the place where the relationships among the meta-models are maintained.

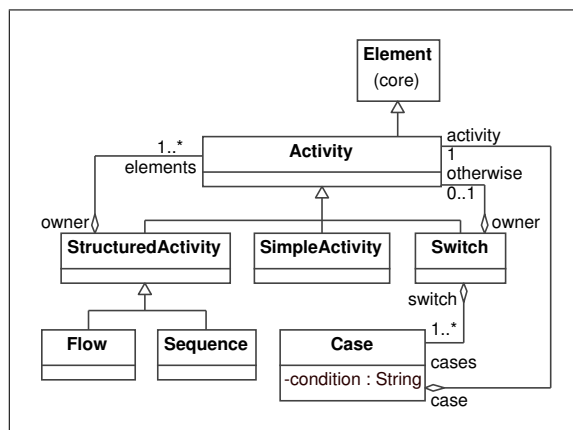


Figure 6: The Orchestration Meta-Model

Orchestration view meta-model Orchestration is one of the most important concerns of a SOA process. An orchestration view as shown in figure 6 comprises many activities and control structures. The activities are process tasks such as service invocations, or data handling, while control structures describe the execution order of the activities to achieve a certain goal.

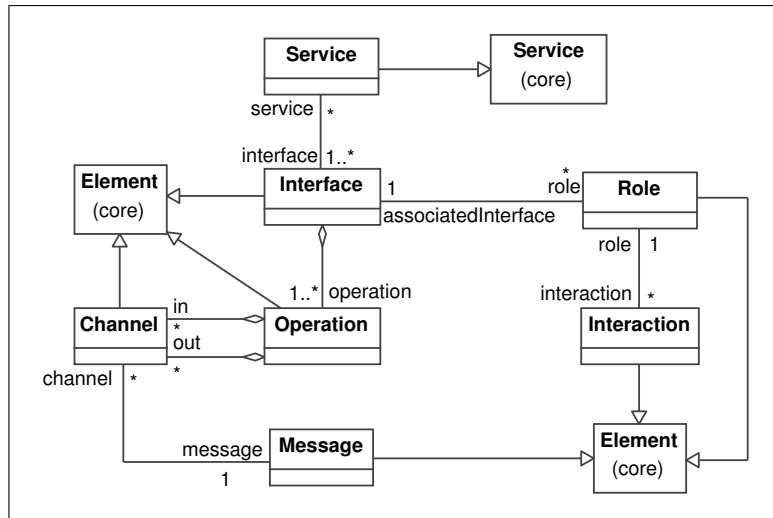


Figure 7: The Collaboration Meta-Model

Collaboration view meta-model The collaboration view meta-model extends the Core meta-model to represent the interactions between the business process and its partners and is shown in figure 7.

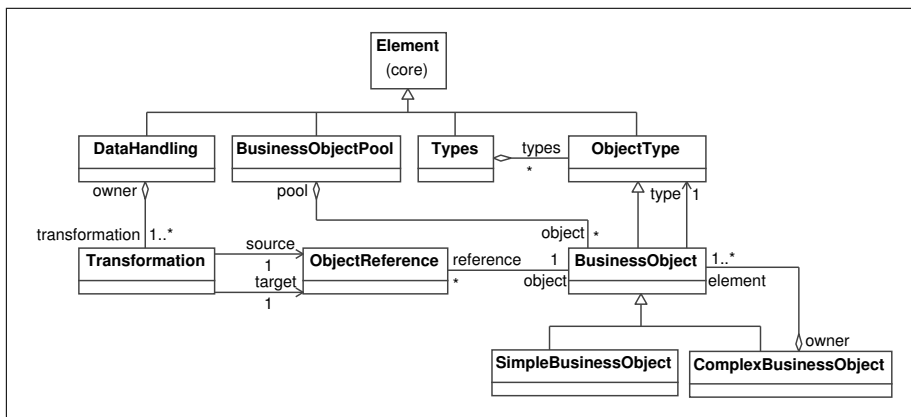


Figure 8: The Information Meta-Model

Information view meta-model An information view is shown in figure 8. This meta-model involves the representation of data object flows inside the process, and message objects traveling back and forth between the process and external components.

4 Conclusion

This deliverable presented the first version of the Business Process Modeling Ontology, together with the methodology followed in developing it. The ontology was developed based on the state of the art and requirements analysis from [14]. As proof of concept, it also present how the product ordering process can be modeled using BPMO.

Several adjustments and updates may be needed, based on further requirements raised by the use cases. This adjustments will be implemented in the next version of the ontology.

A Product Ordering Use Case

In this scenario a customer orders a product through an eTel reseller, the eTel product team, or possibly the web-based shop. The reseller must communicate the order to the eTel provisioning system. The order in this scenario is a bundle of sub products which must be decomposed into the respective provisioning interfaces. At the end of the technical provisioning, the customers service is entered into the billing system.

Listing 12: Product Ordering Ontology

```
wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://www.sembiz.org/etal#",
  bpmo _"http://www.sembiz.org/bpmo#",
  dc _"http://purl.org/dc/elements/1.1#" }

////////////////////////////////////
//////          eTel Product Ordering Use Case          \\\\\\\
////////////////////////////////////

ontology _"http://www.sembiz.org/etal"
  importsOntology
    _"http://www.sembiz.org/bpmo#"

////////////////////////////////////
//////          business roles          \\\\\\\
////////////////////////////////////
concept customer subConceptOf bpmo#Personnel
  nonFunctionalProperties
    dc:description hasValue "order Product bundle"
  endNonFunctionalProperties

concept reseller subConceptOf bpmo#Personnel
  nonFunctionalProperties
    dc:description hasValue "handles customer orders; enters
      product order into eTel system"
  endNonFunctionalProperties

instance crm memberOf bpmo#SoftwareSystem
  nonFunctionalProperties
    dc:description hasValue "stores customer data; initial order"
  endNonFunctionalProperties

instance provisioning memberOf bpmo#SoftwareSystem
  nonFunctionalProperties
    dc:description hasValue "workflow for necessary provisioning
      for order"
  endNonFunctionalProperties

instance billing memberOf bpmo#SoftwareSystem
  nonFunctionalProperties
    dc:description hasValue "repository of entities for customer billing"
  endNonFunctionalProperties

instance cpe memberOf bpmo#Organization
  nonFunctionalProperties
    dc:description hasValue "provides necessary equipments to the
      customer premise for the desired
      service bundle"
  endNonFunctionalProperties

instance telekomAustria memberOf bpmo#Organization
  nonFunctionalProperties
    dc:description hasValue "provides access to PSTN network for voice
```

```

        access; provides connection from DSL
        line to eTel"
    endNonFunctionalProperties
    hasWebsite hasValue "http://www.telekomaustria.com./index-en.php"
    hasLocation ofType "Vienna"

instance eTelISP memberOf bpmo#SoftwareSystem
    nonFunctionalProperties
        dc#description hasValue "eTel ISP Services"
    endNonFunctionalProperties

instance eTelSwitchTeam memberOf bpmo#Organization
    nonFunctionalProperties
        dc#description hasValue "eTel Switch Team"
    endNonFunctionalProperties

concept account subConceptOf bpmo#Asset
    nonFunctionalProperties
        dc#description hasValue "customer account"
    endNonFunctionalProperties
    isactive ofType _boolean

```

Listing 13: Product Ordering

```

wsmlVariant _"http://www.wsmo.org/wsml/wsml-syntax/wsml-flight"
namespace { _"http://www.sembiz.org/etalPO#",
    bpmo _"http://www.sembiz.org/bpmo#",
    dc _"http://purl.org/dc/elements/1.1#",
    etel _"http://www.sembiz.org/etal#" }

////////////////////////////////////
//////          Product Ordering          \\\\\\\
////////////////////////////////////

ontology _"http://www.sembiz.org/etal/po"

////////////////////////////////////
//////          business role            \\\\\\\
////////////////////////////////////
instance _customer1 memberOf etel#customer

instance _reseller1 memberOf etel#reseller

instance _equipmentReseller memberOf etel#reseller

////////////////////////////////////
//////          business data: events    \\\\\\\
////////////////////////////////////
instance _start memberOf bpmo#Event
    nonFunctionalProperties
        dc#description hasValue "customer wants a service bundle"
    endNonFunctionalProperties

instance _orderCreated memberOf bpmo#Event
    nonFunctionalProperties
        dc#description hasValue "order created in ordering system"
    endNonFunctionalProperties

instance _DSLInitated memberOf bpmo#Event
    nonFunctionalProperties
        dc#description hasValue "DSL provisioning initiated"
    endNonFunctionalProperties

instance _IPInitated memberOf bpmo#Event

```

```

        nonFunctionalProperties
          dc:description hasValue "IP services provisioning initiated"
        endNonFunctionalProperties

instance _voiceInitiated memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "Voice Provisioning initiated"
  endNonFunctionalProperties

instance _EquipmentConfigured memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "Equipment configured"
  endNonFunctionalProperties

instance _DSLActive memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "DSL line becomes active"
  endNonFunctionalProperties

instance _DNSAdminETel memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "DNS domain now administered by eTel"
  endNonFunctionalProperties

instance _voiceEnabled memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "Voice circuit enabled"
  endNonFunctionalProperties

instance _EmailProvisioned memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "Email accounts provisioned"
  endNonFunctionalProperties

instance _ProductCompleted memberOf bpmo#Event
  nonFunctionalProperties
    dc:description hasValue "Product Completed"
  endNonFunctionalProperties

////////////////////////////////////
//////          business process
////////////////////////////////////

////////// the nine atomic process in the eTel Product Ordering use case
instance _NegotiationBundle memberOf bpmo#AtomicProcess
  nonFunctionalProperties
    dc:description hasValue "negotiation of the exact details
      of the service bundle"
  endNonFunctionalProperties
  bpmo#withExecution hasValue _execution1
  bpmo#withCapability hasValue _capability1
instance _capability1 memberOf bpmo#Capability
  bpmo#hasInput hasValue _start
  bpmo#hasOutput hasValue _orderCreated
instance _execution1 memberOf bpmo#Execution
  bpmo#executedBy hasValue _reseller1

instance _StartProvisioning memberOf bpmo#AtomicProcess
  nonFunctionalProperties
    dc:description hasValue "start the provisioning steps"
  endNonFunctionalProperties
  bpmo#withExecution hasValue _execution2
  bpmo#withCapability hasValue _capability2
instance _capability2 memberOf bpmo#Capability
  bpmo#hasInput hasValue _orderCreated

```

```

    bpmo#hasOutput hasValue _DSLInitated
    bpmo#hasOutput hasValue _IPInitated
    bpmo#hasOutput hasValue _voiceInitated
instance _execution2 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#provisioning
    bpmo#hasPreProcess hasValue _NegotiationBundle

instance _CustomerPremise memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "customer premise equipment ordered"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution3
    bpmo#withCapability hasValue _capability3
instance _capability3 memberOf bpmo#Capability
    bpmo#hasInput hasValue _DSLInitated
    bpmo#hasOutput hasValue _EquipmentConfigured
instance _execution3 memberOf bpmo#Execution
    bpmo#executedBy hasValue _equipmentReseller
    bpmo#hasPreProcess hasValue _StartProvisioning

instance _DSLOrderPlaced memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "DSL order placed with Telecom Austria"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution4
    bpmo#withCapability hasValue _capability4
instance _capability4 memberOf bpmo#Capability
    bpmo#hasInput hasValue _DSLInitated
    bpmo#hasOutput hasValue _DSLActive
instance _execution4 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#telekomAustria
    bpmo#hasPreProcess hasValue _StartProvisioning

instance _DNSRegister memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "DNS domain registration"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution5
    bpmo#withCapability hasValue _capability5
instance _capability5 memberOf bpmo#Capability
    bpmo#hasInput hasValue _IPInitated
    bpmo#hasOutput hasValue _DNSAdminETel
instance _execution5 memberOf bpmo#Execution
    bpmo#hasPreProcess hasValue _StartProvisioning

instance _EmailConfiguration memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "Email accounts configuration for DNS domain"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution6
    bpmo#withCapability hasValue _capability6
instance _capability6 memberOf bpmo#Capability
    bpmo#hasInput hasValue _DNSAdminETel
    bpmo#hasOutput hasValue _EmailProvisioned
instance _execution6 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#eTelISP
    bpmo#hasPreProcess hasValue _DNSRegister

instance _VoicePreSelection memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "Voice Carrier pre-selection"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution7

```

```

    bpmo#withCapability hasValue _capability7
instance _capability7 memberOf bpmo#Capability
    bpmo#hasInput hasValue _voiceInitated
    bpmo#hasOutput hasValue _voiceEnabled
instance _execution7 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#telekomAustria
    bpmo#hasPreProcess hasValue _StartProvisioning

instance _eTelSwitch memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "eTel switch provisioning"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution8
    bpmo#withCapability hasValue _capability8
instance _capability8 memberOf bpmo#Capability
    bpmo#hasInput hasValue _voiceInitated
    bpmo#hasOutput hasValue _voiceEnabled
instance _execution8 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#eTelSwitchTeam
    bpmo#hasPreProcess hasValue _StartProvisioning

instance _AccountEntered memberOf bpmo#AtomicProcess
    nonFunctionalProperties
        dc#description hasValue "Account entered into billing system"
    endNonFunctionalProperties
    bpmo#withExecution hasValue _execution9
    bpmo#withCapability hasValue _capability9
instance _capability9 memberOf bpmo#Capability
    bpmo#hasInput hasValue _EquipmentConfigured
    bpmo#hasInput hasValue _DSLActive
    bpmo#hasInput hasValue _EmailProvisioned
    bpmo#hasInput hasValue _voiceEnabled
    bpmo#hasOutput hasValue _ProductCompleted
instance _execution9 memberOf bpmo#Execution
    bpmo#executedBy hasValue etel#billing
    bpmo#hasPreProcess hasValue _CustomerPremise
    bpmo#hasPreProcess hasValue _DSLOrderPlaced
    bpmo#hasPreProcess hasValue _EmailConfiguration
    bpmo#hasPreProcess hasValue _VoicePreSelection
    bpmo#hasPreProcess hasValue _eTelSwitch

////////// the composite processes in the eTel Product Ordering use case
instance _DSLCompositeProcess memberOf bpmo#ParallelProcess
    nonFunctionalProperties
        dc#description hasValue "DSL Provisoning Process"
    endNonFunctionalProperties
    bpmo#subprocess hasValue _CusterPremise
    bpmo#subprocess hasValue _DSLOrderPlaced

instance _IPCompositeProcess memberOf bpmo#SequenceProcess
    nonFunctionalProperties
        dc#description hasValue "IP Services Provisoning Process"
    endNonFunctionalProperties
    bpmo#headProcess hasValue _DNSRegister
    bpmo#tailProcess hasValue _EmailConfiguration

instance _VoiceCompositeProcess memberOf bpmo#ParallelProcess
    nonFunctionalProperties
        dc#description hasValue "Voice Provisoning Process"
    endNonFunctionalProperties
    bpmo#subprocess hasValue _VoicePreSelection
    bpmo#subprocess hasValue _eTelSwitch

// The Provisoning Process
instance _ThreeProvisoningProcess memberOf bpmo#ParallelProcess
    nonFunctionalProperties

```

```

        dc:description hasValue "Three Provisioning Process"
    endNonFunctionalProperties
    bpmo#subprocess hasValue _DSLCompositeProcess
    bpmo#subprocess hasValue _IPCompositeProcess
    bpmo#subprocess hasValue _VoiceCompositeProcess

// the whole PO Process
instance _VoiceCompositeProcess memberOf bpmo#SequenceProcess
    nonFunctionalProperties
        dc:description hasValue "Product Ordering Process"
    endNonFunctionalProperties
    bpmo#headProcess hasValue _NegotiationBundle
    bpmo#tailProcess hasValue _VoiceCompositeProcess2
instance _VoiceCompositeProcess2 memberOf bpmo#SequenceProcess
    bpmo#headProcess hasValue _StartProvisioning
    bpmo#tailProcess hasValue _VoiceCompositeProcess3
instance _VoiceCompositeProcess3 memberOf bpmo#SequenceProcess
    bpmo#headProcess hasValue _ThreeProvisioningProcess
    bpmo#tailProcess hasValue _AccountEntered

```

References

- [1] W. Aalst. Formalization and Verification of Event-driven Process Chains. Computing Science Reports 98/01, Eindhoven University of Technology, Eindhoven, 1998.
- [2] A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. Knig, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M. Zeller. Web Services Human Task, (WS-HumanTask), Version 1.0. available at: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf, June 2007.
- [3] A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. Knig, F. Leymann, R. Müller, G. Pfau, K. Plösser, R. Rangaswamy, A. Rickayzen, M. Rowley, P. Schmidt, I. Trickovic, A. Yiu, and M. Zeller. WS-BPEL Extension for People, (BPEL4People), Version 1.0. available at: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf, June 2007.
- [4] P. Biron and A. Malhotra. XML Schema Part 2: Datatypes Second Edition. W3C Recommendation, October 2004.
- [5] E. P. Bontas. *A Contextual Approach to Ontology Reuse: Methodology, Methods and Tools for the Semantic Web*. PhD thesis, Freien Universität Berlin, 2006.
- [6] J. de Bruijn, H. Lausen, R. Krummenacher, A. Polleres, L. Predoiu, M. Kifer, and D. Fensel. The Web Service Modeling Language WSML. WSML Working Draft D16.1v0.21, available at: <http://www.wsmo.org/TR/d16/d16.1/v0.2/>, October 2005.
- [7] 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.
- [8] Eclipse. Eclipse modeling framework. <http://www.eclipse.org/emf/>, 2006.

- [9] M. Evenson and B. Schreder. D4.1 Use Case Definition and Functional Requirements Analysis. SemBiz Deliverable, August 2007.
- [10] M. Kerrigan, A. Mocan, M. Tanler, and D. Fensel. The Web Service Modeling Toolkit - An integrated development environment for semantic web services. In *Proceedings of the 4th European Semantic Web Conference (ESWC), System Description Track*, Innsbruck, Austria, June 2007.
- [11] A. Mocan and E. Cimpian. An ontology-based data mediation framework for semantic environments. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3(3), 2007.
- [12] 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.
- [13] H. Tran, U. Zdun, and S. Dustdar. View-based and model-driven approach for reducing the development complexity in process-driven soa. In *Int. Conf. on Business Process and Services Computing (BPSC)*, Leipzig, Germany, Sept. 2007.
- [14] Z. Yan, E. Cimpian, M. Mazzara, and M. Zaremba. D1.1 BPMO Requirements Analysis and Design. SemBiz Deliverable, February 2007.