



Semantic Annotations for Workflow Interoperability

Hamri Salah, Boudjlida Nacer, Boufaïda Mahmoud

► To cite this version:

Hamri Salah, Boudjlida Nacer, Boufaïda Mahmoud. Semantic Annotations for Workflow Interoperability. *Informatica*, 2014, 38 (4), pp.347-366. hal-01111453

HAL Id: hal-01111453

<https://inria.hal.science/hal-01111453>

Submitted on 30 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Semantic Annotations for Workflow Interoperability

Hamri Salah
Laboratory LIRE, University of Constantine 2, Algeria
hamri.salah@yahoo.fr

Boudjlida Nacer
University of Nancy 1, LORIA
UHP, France
Nacer.Boudjlida@loria.fr

Boufaida Mahmoud
Laboratory LIRE, University of Constantine 2, Algeria
mboufaida@umc.edu.dz

Keywords: Workflow Model, Interoperability, Semantic Annotations, Ontologies, Common Workflow Ontology, Common Process Semantic Annotation Model.

Received:

Abstract. To support the interoperability or the cooperation between different partners, various approaches and technological solutions were proposed, which converge directly to the adoption of standards. Consequently, the semantic aspect is not correctly addressed by today's interoperability solutions that focus mainly on the syntactical and technical level. Indeed, addressing the semantic aspect at conceptual level will provide more flexibility to the cooperation. Accordingly, in this paper, we propose an agnostic approach for the interoperability of Workflow models (or business process), which is used in a homogeneous or in a heterogeneous context. In a homogeneous context, lexical and structural annotations are attached to models. Contrary, in a heterogeneous context, we introduce a common semantic annotation structure for annotating the models at different levels: 1) meta-models, 2) models content, 3) models profiles and goals for semantic discovery purposes, 4) at levels of basic aspects of models such as the informational type. Common ontologies, including: Workflow ontology, domain specific ontology, profiles ontology, goals ontology and a set of ontologies related to these aspects, are used to achieve semantic interoperability. One of the advantages of this proposal is its flexibility and its openness since we take an agnostic approach to ontology representation languages (such as OWL-S or WSDL-S).

Povzetek:

1 Introduction

To enable interoperability between Workflow models, the WfMC (Workflow Management Coalition) defined a canonical model [8] called XPDL (XML - Process Language Definition) as a language for process interchange and Wf-XML [9] as interoperability protocol between Workflow engines. The Business Process Management (BPM) have mainly focused on Web service technology and came up with a multitude of Web service composition languages standards such as ebXML [3], BPEL4WS (Business Process Execution Language for Web Services) [7]. However, these languages are based on XML and unfortunately lack a semantic description of concepts that enable business process to exchange with a common understanding.

To address the need of semantics, different languages have been proposed such as OWL-S (Ontology Web Language for services) [10], WSMO (Web Service

Modeling Ontology) [11], among others. Unfortunately, the problem of the interoperability ever remains at technical level using these ontology representation languages.

To fulfil the need of an independent way of describing semantics to Web services, some authors [13], [14], [15], [16] have developed an approach based on MDA (Model Driven Architecture) [6] for generating OWL-S descriptions. As such, their approach relies upon the use of the OWL-S language, whereas in [4], the authors developed an approach that focuses on WSMO language. Thus, the semantic aspect is not correctly addressed in a conceptual manner.

For bringing semantics to process models at conceptual level (without looking any technology and associated languages), some authors [17] [18], [19], [20], [39], [34]

have proposed the use of the semantic annotations by adding metadata and using a set of ontologies to describe the semantics of information in a heterogeneity context. However, their approach doesn't consider the model aspects while they are essential and intrinsically related to any process model such as the informational or the organizational type. As argued in [43], the purpose of a Workflow model (or business process) is to be executed. Therefore, it is natural for us to refer to its different aspects for execution and to its main objective for which it has been created.

In addition to these works, many approaches have been conducted, with different points by researchers, such as [45], [46], [47], [48], [49] leading to various forms of semantic annotations for process models. However, the authors deal with the problem of semantic heterogeneity of models independently of the context of interoperability where we should align the different terminology that might be used in models and share a conceptualization of the concepts typically employed in the most process meta-models (or process modeling languages). There is no common Workflow ontology (or common process ontology) that enables a common understanding of the models between the actors that wish to interoperate. Additionally, the model aspects are not considered in these approaches.

Moreover, the authors have disregarded the homogeneous context where sometimes the interoperating actors find difficulties for better interpreting their models. These difficulties are caused in general by the unambiguous terms used in the Workflow domain (or business process area).

Consequently, to enhance semantic interoperability in a homogeneous context, we introduce complementary annotations helpful the cooperating actors for expliciting the meaning of the models and for easing their exchanged within that context.

This motivates our interest in developing two conceptual approaches that allow actors to cooperate in both homogeneous and heterogeneous environments.

In the first approach, we suggest the use of lexical and structural annotations for annotating the models within a homogeneous context. Then, in the second approach, to tackle the heterogeneous semantics of distributed process models, we propose an ontology based semantic annotation approach based on building a common process semantic annotation model (CPSAM) for annotating the models at different levels: 1) meta-models, 2) models content, 3) models profiles and goals for semantic discovery purposes, 4) models aspects such as the informational or the organizational type.

The contribution of this paper is as follows. First, we propose an approach for Workflow interoperability that enables the cooperating actors to better interpret the meaning of the exchanged models in a homogeneous

context. Lexical and structural annotations are used within that context for annotating the models. Second, to achieve semantic interoperability in a heterogeneity context, a common process semantic annotation model is developed and presented formally. This explicitly defines all necessary annotation elements. We introduce in that context, a set of common ontologies, including: Workflow ontology, domain specific ontology, etc. Third, we define the mapping rules for the meta-models and the model annotation.

The remainder of this paper is structured as follows. In the next section, we describe related work. Then, to achieve semantic interoperability of Workflow models in a homogeneous and heterogeneous context, we propose in Section 3, the different types of annotations that can be attached to models. In section 4, we build a common process semantic annotation model (CPSAM) whose concepts are extracted from common and shared ontological concepts. Section 5 introduces the Supply Chain Operations Reference (SCOR) [36] model as example of reference domain ontology for annotating the models content. Then, we formalize CPSAM in Section 6. For semantic discovery purposes of these models, we enrich in Section 7, the CPSAM with semantic annotations of profiles and goals. Section 8 complements the CPSAM with another type of annotations, which is related to basic aspects of models. Finally, we conclude this paper and outline our future work.

2 Related Work

The Web services community has proposed different Web services composition languages such as BPEL4WS [7], WSFL (Web Service Flow Language) [5], ebXML [3] for the interoperability of business process. However, these languages lack a semantic description of concepts for a common understanding of models.

To provide semantic descriptions to Web services, some authors have investigated the area of semantic Web services and have proposed different languages such as OWL-S (Ontology Web Language for services) [10], WSMO (Web Service Modelling Ontology) [11], WSDL-S (Web Service Description Language Semantics) [12], among others. However, the authors rely on the use of these ontology languages to describe the semantics of Web services. Thus, they deal with the semantic aspect at technical level using technologies (tools, existing platforms, languages, etc.) for implementing their approach.

To solve the problem of the semantics of Web services in any independent way of these ontology representation languages, some authors [13], [14], [15], [16] have proposed to create OWL-S descriptions using the MDA approach. Their purpose is to allow a developer to focus on creation of semantic Web services and associated OWL-S specifications via the development of a standard UML model. By using MDA approach, the technique facilitates the creation of descriptions of semantic

concepts while hiding the syntactic details associated with creating OWL-S specifications. As it was shown previously in [1], [2], we have proposed a way of addressing the challenges of semantic description of Workflow models (or process models) using the MDA standards and in particular the Ontology Definition Meta-model (ODM) for ontology modeling. However, such approaches are usually applied in an industrial context. Hence, the problem of the semantics of Web services is not correctly addressed in a conceptual way.

In business process area, to specify the semantics of process models, some authors [17] [18], [19], [20], [39], [34] have proposed the use of semantic annotations for achieving semantic interoperability at conceptual level. Their approach focuses on the semantic heterogeneity of the process models on both model and meta-model levels. We see our work as an extension to the existing efforts, because the approach discusses these efforts in the context of interoperability of business process from the perspective of model and meta-model levels. However, the authors omit the basic aspects, which are related to models for execution. Furthermore, they don't consider the homogenous context where the cooperating actors need sometimes additional annotations helpful for a common interpretation of the models within that context.

Similarly, there are other approaches of semantic interoperability such as [45], [46], [47], [48], [49]. Their purpose is to provide semantic annotations for process flows, as well as implemented research prototypes with similar or overlapping capabilities. However, the authors are unaware of context of interoperability where we need a common Workflow ontology (or common process ontology) that should be shared between the actors that wish to interoperate. Furthermore, the authors generally deal with semantic heterogeneity on modeling language level, and omit the meta-model level, which is essential in a context of interoperability. Indeed, within that context, we should address the semantic heterogeneity problem not only on model level or on language such as PSL, but on meta-model level and use one or several ontologies that provide a representation of conceptualization of concepts typically used in the most process modeling languages. Thus, we can facilitate interoperability.

The common factor of these approaches is omitting the most important aspects of business processes, which can be modeled and be investigated independently of each other. As argued in [43], the purpose of any Workflow model or business process is to be executed.

Thus, we must take into account these aspects. Among them, we cite those which are generally considered as essential such as the informational, the organizational, the functional, the behavioural or the resources aspect.

Also, recent works [40], [41] [42], affirm that annotating business processes with a context-based process semantic

annotation model facilitates searching models, navigating the repository and enhance understandability of process models. The annotation model consists of the following annotation elements: *process type*, *process area*, *resource*, *actor*, *organizational level*, *process phase*, *process relationship*, *business context*, and *goal*.

However, this annotation model is composed of elements derived from concepts which were elicited (from literature) and validated through an empirical study. Hence, there is no formal semantics of concepts and no reference to one or several ontologies. Indeed, having ontologies will provide a representation of a shared conceptualization and a common understanding of process models in concise and consensual manners for the cooperating partners. Thus, the approach we adopt is based on ontologies and semantic metadata as mediators for reconciling the semantic heterogeneity of process modeling concepts.

Finally, a look at these current approaches has shown, that is a large emphasis on a heterogeneity context and less work in homogeneity context. Indeed, in a real business cooperation and in a homogenous context, the collaborating actors need sometimes annotations when difficulties arise in the interpretation and the understanding of the models.

Accordingly, we address in this paper the semantic heterogeneity problem on both homogenous and heterogeneous contexts. We use ontologies to relate concepts across different modeling languages, as well as to align domain specific terminology used in modeling languages.

Our purpose is therefore to propose a conceptual semantic annotation framework for interoperability of Workflow models (or business process) within or across enterprises. For this purpose, we propose two approaches, which are used in a homogeneous and in a heterogeneous context. In a homogenous context, we attached to models: 1) lexical annotations with reference to lexical data base, 2) structural annotations using only the UML notation. Contrary, in a heterogeneous context, the proposed approach is based on building a common process semantic annotation model (CPSAM) that contains all the common concepts that are shared between the actors that wish to interoperate. In that context, we refer to a set of common ontologies for annotating the models at levels of meta-models and models. To enhance interoperability and to enable a common interpretation and understanding of the models without any ambiguity; we extend CPSAM with annotations of basic aspects of models (such as the informational or the organizational aspect). To enhance reuse of models in their specific projects and especially for profiles or goals-oriented queries, we complement CPSAM with profiles and goals annotations.

One of the advantages of this proposal is its flexibility and its openness since we take an agnostic approach to

ontology representation languages (such as OWL-S or WSDL-S). In this way, the cooperating actors can annotate their models (or models fragments) according to their preferred ontology representation language.

3 Annotations for workflow interoperability

In this section, we propose the types of annotations that we feel necessary to achieve semantic interoperability of Workflow (or process) models in a conceptual way. However, two contexts can occur between two interoperating actors: *i*) within a homogenous context in intra-enterprise cooperation, *ii*) within a heterogeneous context in inter-enterprise cooperation.

3.1 Annotations in a homogeneous context

The semantic interoperability within a homogeneous environment does not really constitute a crucial problem for the cooperative companies. Nevertheless, to avoid ambiguities of interpretations and having a good comprehension of the models during the cooperation, it is preferable to annotate them.

In a homogenous context, fitting the models with lexical and structural annotations using the UML notation with reference to same ontologies are sufficient for the actors to better interpret the received models.

In this environment, the cooperative partners (or actors) know each other, share the same field of knowledge, (i.e., even area of reference). Therefore, they must agree on the semantic description of the annotations as well as the various types of annotations, which can be thus elaborate together. Therefore, the reference domain ontology is known and two actors exchange their models *M1* and *M2*, using the same notations for their models as well as the same process ontology *O1* or *O2*. This means that *MM1* and *O1* are same as *MM2* and *O2*, respectively or, equivalently. In this context, lexical annotations described by an actor *Act1* for the model *M1* suffice for the actor *Act2* to interpret the received model. Moreover, to enable consistently understand the meaning of models basic aspects without ambiguity, the interoperating actors refer to same ontologies corresponding to each aspect.

In this homogeneous context, we propose two types of annotations:

- *Lexical/Terminological annotations:* By such annotation, terms are attached to a lexical base data, which make explicit the meaning of models. The definition of terms may be provided by the available ontologies, by the modeling language, as well as by WordNet [21] or by Wikipedia [22]. In addition to this kind annotation, we use a common glossary [23],

which is established by the WfMC (Workflow Management Coalition) [24] and contains a description and common terminology for the basic concepts embodied within a process definition.

- *Structural annotations:* that consist in attached concepts from domain ontology to models elements such operations and attributes using the UML notation. The figure 1 shows an example of structural annotations of a purchase order shipping activity (a model fragment) where the elements of the activity refer to a set of ontologies (purchase order, shipment, bank and finances).

Although, we are in a homogeneous context (i.e., same domain ontology (for instance, an automobile area)), we can distinguish two situations that can occur between two actors that wish to interoperate within that context:

Situation 1: When the interoperating actors *Act1* and *Act2* use the same ontologies: This means that is a same understanding that may rely on the use of one or several ontologies, which are identical. These ontologies concern the process models and their basic aspects. In this situation, the annotator annotates the process models *M1* and *M2*, using only the lexical annotations that express the natural meaning of terms. The definitions of the terms may be provided by some lexicons or agreed terminology (like those by WordNet [21], by Wikipedia [22] or by the WfMC's glossary [23], which is established by the organization Workflow [24]).

Situation 2: When the interoperating actors *Act1* and *Act2* don't agree on the same ontologies: This means that each actor use its proper ontology. Thus, *Act1* annotates his model with reference to the ontology *O2*, which is used by *Act2*, i.e., the *O2* concepts are used as metadata to annotate the model *M1*. Also, *Act2* uses the *O1* concepts as metadata to annotate the model *M2*.

However, in a heterogeneous context, semantics interoperability constitutes a difficult problem in meaning understanding and sharing models in inter-enterprise cooperation. Therefore, we must deal with the semantic heterogeneity of Workflow meta-models and models within that context.

3.2 Annotations in a heterogeneous context

In a heterogeneous context, the cooperating actors may use different notations for their models (for example, *M1* is notated according to the meta-model *MM1* and *M2* to the meta-model *MM2*) and they may refer to different ontologies. For example, *Act1* uses an ontology *O1*, while *Act2* uses an ontology *O2* different from *O1*.

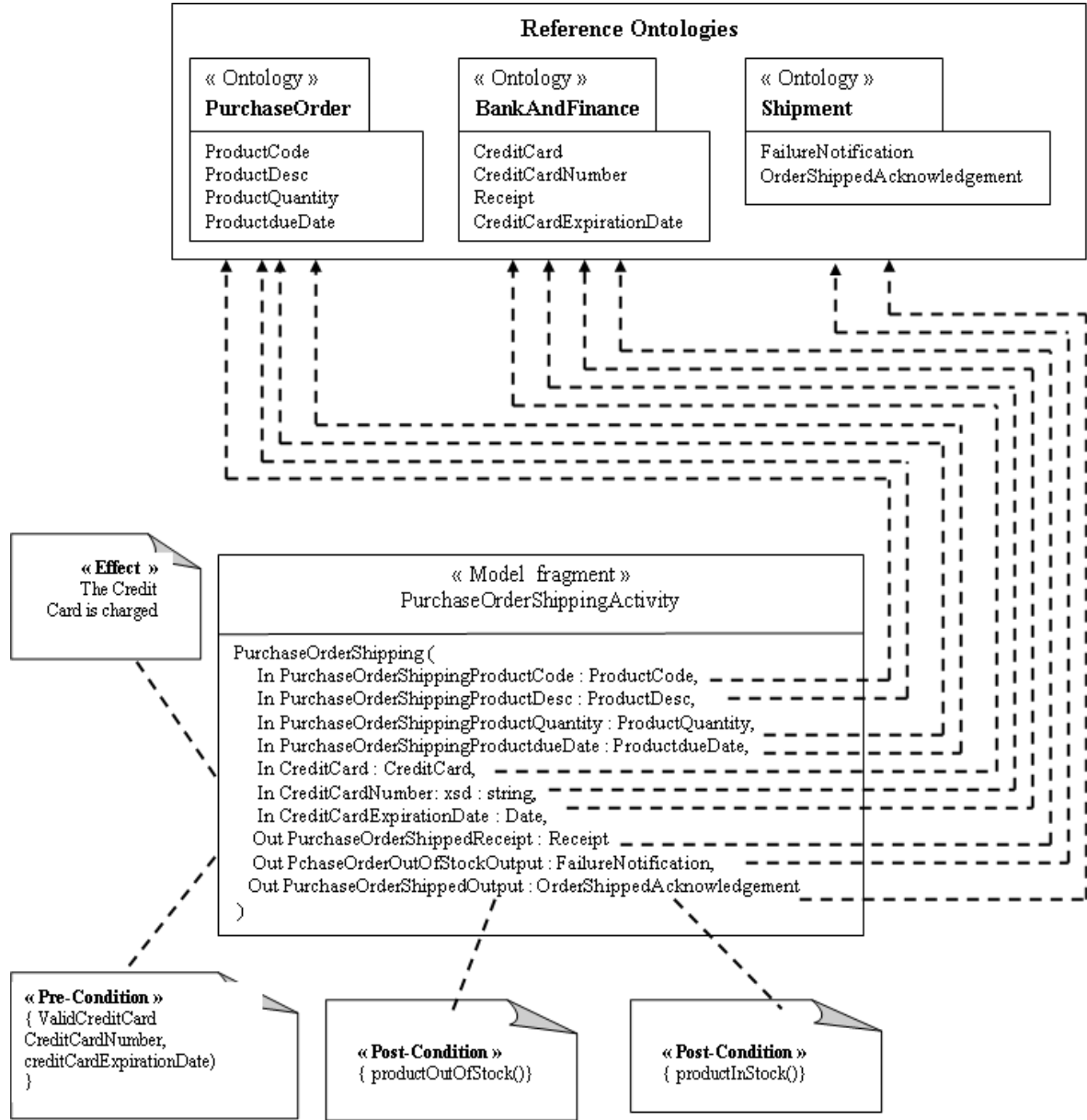


Figure 1: Structural annotations of a model fragment (purchase order-shipping activity).

In order to achieve semantic interoperability of process models within that context, a common understanding of models representation is needed when searching process models. Therefore, we use common ontologies to relate concepts across different modeling languages, as well as to align domain specific terminology used in models. Thus, we annotate the models with reference to a set of common ontologies that are consensually, accept and shared between the interoperating actors.

The proposed approach is therefore an ontology based semantic annotation approach. It relies on the use of several common ontologies, including: Workflow ontology, domain specific ontology, profiles ontology, goals ontology and a set of ontologies related to different basic aspects, which are intrinsically related to process models such as the informational type. For annotating the

various models, we define a semantic annotation process (figure 2) based on the following steps:

1. Elaboration of a common process semantic annotation model (CPSAM): In this step, we build a common process semantic annotation model whose concepts derived from a common Workflow ontology (CWO), reference domain ontology and a thesaurus (as a form of ontology).
2. Meta-model annotation: It consists in annotating the concepts of different meta-models of Workflow (such as XPD or ebXML) by the concepts of the common Workflow Ontology (CWO). The CWO concepts are then used as metadata to annotate the various semantics of concepts of meta-models, i.e. the CWO

concepts will take place the corresponding meta-models concepts to describe the processes.

we annotate them with reference to profiles and goals ontology.

3. Model annotation: Once, the concepts of meta-models are annotated by the CWO concepts, (i.e. the process models are described by the CWO concepts), we annotate the models content with reference to a domain specific and thesaurus.
4. Profiles and goals annotation: For semantic discovery purposes and especially for profiles or goals-oriented queries, and reuse models in their specific projects,
5. Annotation of basic aspects: For better interpreting the meaning of the exchanged models without ambiguity, we annotate the models with reference to a set of common ontologies, which are related to the informational, the organizational, the behavioural, the functional or the resources aspect.

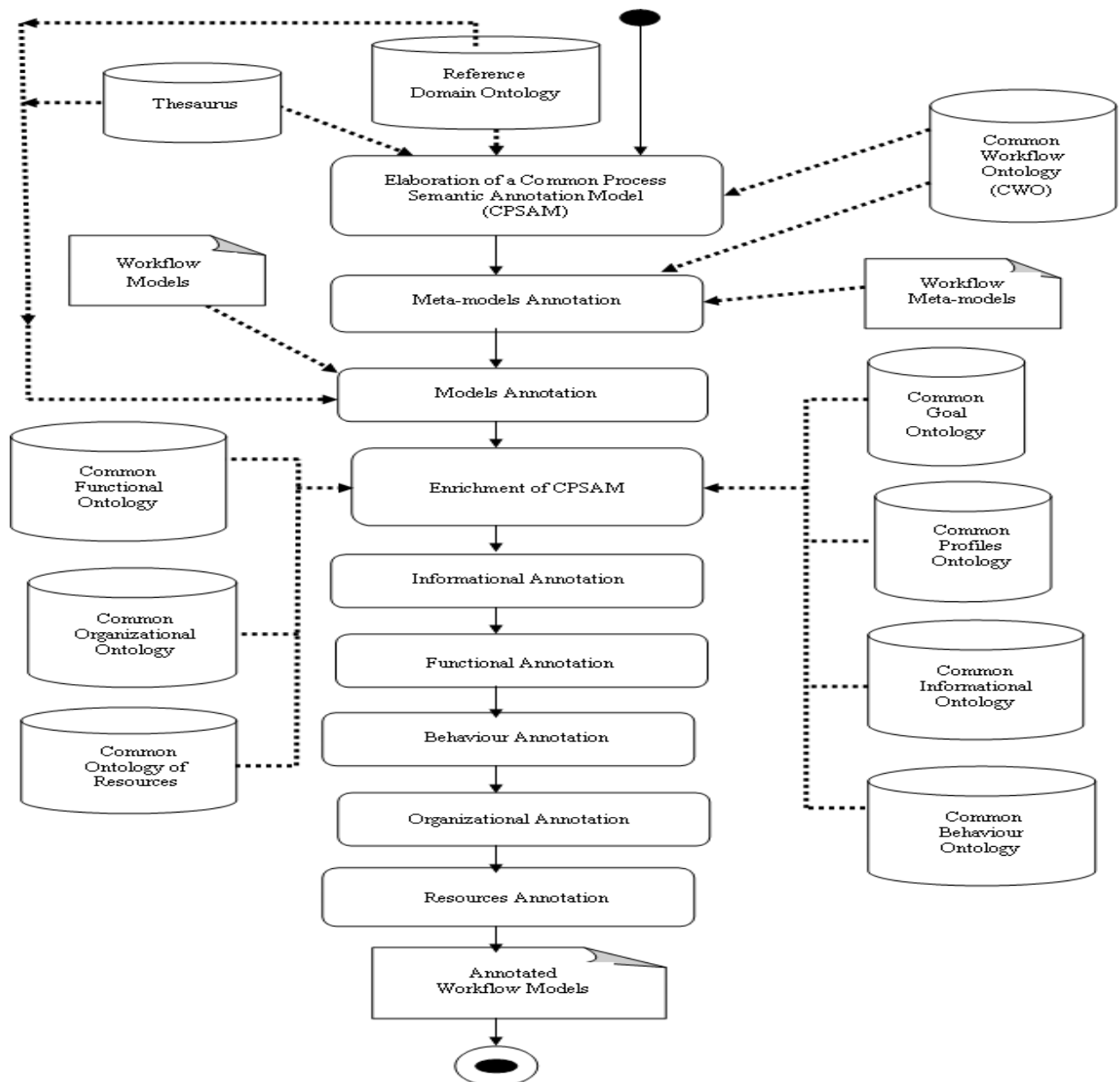


Figure 2: Semantic Annotation Process of Workflow Models.

4 Common process semantic annotation model

To support the various semantic annotations, we define a common annotation scheme (figure 3) dedicated to actors that wish to cooperate. This scheme represents a common process semantic annotation model (CPSAM) that contains all the common and shared ontological concepts between the interoperating actors. These concepts are extracted from a set of common ontologies: Workflow ontology, domain specific ontology, etc.

Following our semantic annotation process as described above, we firstly annotate the semantics of meta-models of process modeling languages. Next, we annotate the

models content, then their profiles, their goals and finally, their basic aspects.

4.1 Meta-model annotation

In the meta-model annotation, we use common Workflow ontology (CWO) as metadata to annotate the semantics of concepts of the different meta-models used in the Workflow domain (or business process). Therefore, we first introduce our common Workflow ontology, and then we describe the way of annotating the semantics of meta-models of process modeling languages.

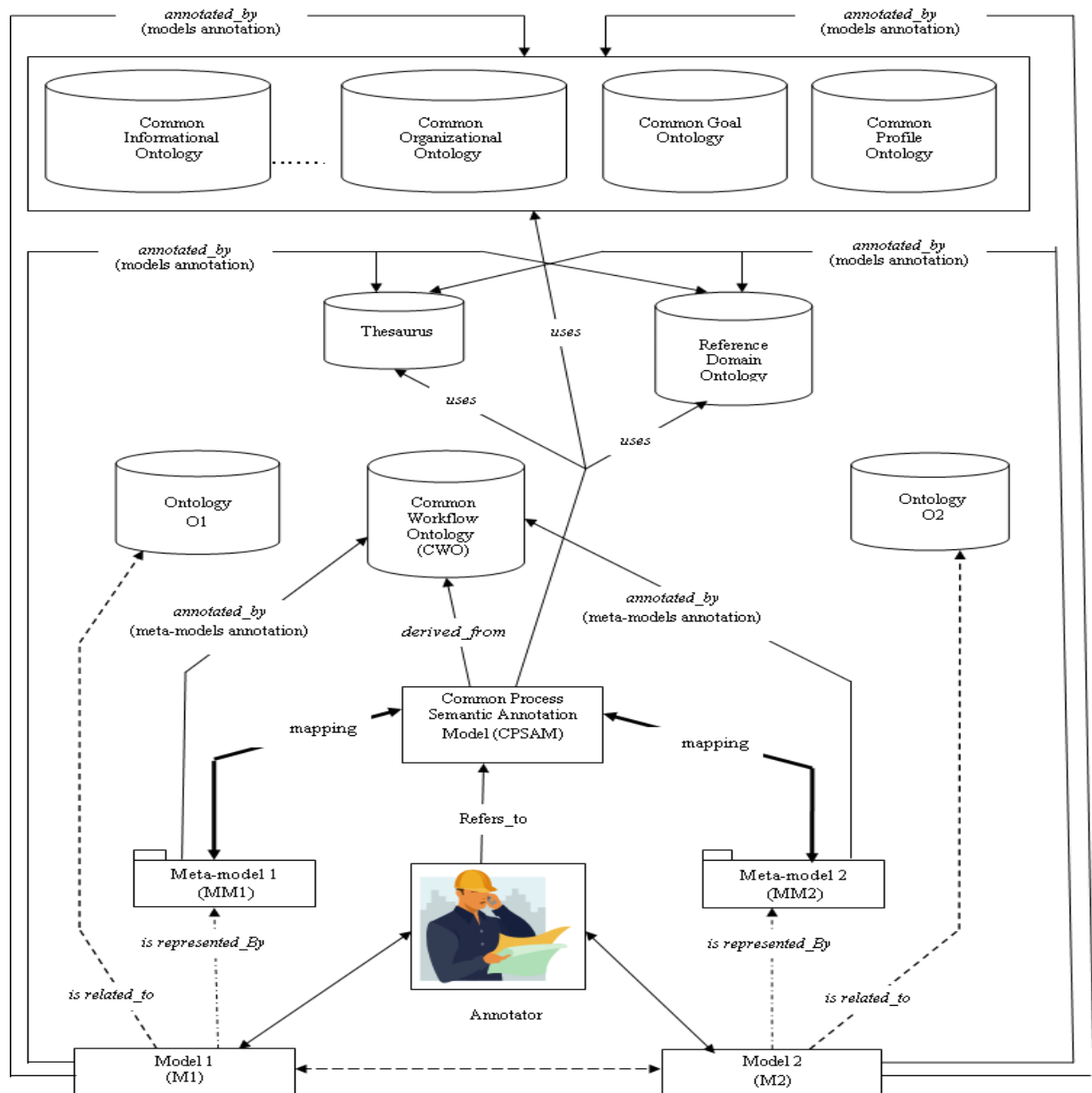


Figure 3: Common Annotation Scheme.

4.1.1 Common workflow ontology

The common Workflow ontology (CWO) is used to align the heterogeneous meta-models of process models. CWO is implemented as an ontology using the Protégé tool [25]. In previous works [1], [2], we have developed an MDA approach for building an OWL ontology for Workflow models according to the investigation of some process modeling languages including ebXML, WSFL, XLANG, BPML, UML, XPD. We have compared the proposed concepts and have aligned them up according to their objectives as defined by their designers. Then, we have extracted a core of basic concepts that is common to the set of Workflow models. Table 1 shows the alignment of concepts of some process meta-models.

The common Workflow ontology is then derived from a common meta-model (figure 4) that includes the following concepts which are usually modeled as modelling constructs in most process modeling languages : *Wf-Process*, *Wf-Step*, *Wf-Activity*, *Wf-Task*, *Wf-Transition*, *Wf-Resource*, *Wf-Artifact*, *Wf-Role*, *Wf-ManualTask*, *Wf-AutomatedTask*. Compared with our previous works [1], [2], this meta-model is updated here by refining *i)* the concept of *Wf-Activity* with the relationships: ‘kind_of’, and ‘phase_of’ *ii)* the concept of *Wf-Actor* with the relationships: ‘member_of’, ‘subClass_of’ and ‘instance_of’ *iii)* the concept of *Wf-Role* with the relationships: ‘member_of’, subClass_of’ and ‘instance_of’. These relationships are used to link the

concepts in models and concepts in ontologies for the model annotation.

4.1.2 Mapping rules in meta-model annotation

The annotation of certain meta-model has to be done manually by experts who know the process modeling language to be annotated. The procedure of a meta-model annotation is in fact to set mapping rules between the CWO concepts and process modeling language constructs or meta-model concepts.

The mapping rules consist of both one-to-one and one to many correspondences between CWO and meta-model concepts. There may be more complicated cases: a correspondence between a CWO concept and a combination of some meta-model concepts. To define the mapping rules for different cases, we categorize only two of modeling constructs: Atomic Concept, Enumerated Concept. Each concept is an Atomic Concept or an Enumerated Concept. This one is an enumeration set of several Atomic Concepts.

Mapping rules

To establish the correspondence between the concepts of meta-models and CWO concepts, we define two types of mapping:

Workflow Concepts	Concepts of Common Meta-Model (CMM)	XPDL Concepts	ebXML Concepts	WSFL Concepts	XLANG Concepts	BPML Concepts	UML Concepts
Process	Wf-Process	Workflow Process	Multiparty Collaboration Binary Collaboration	FlowModel	Process	Process Abstract	ActivityGraph
Activity	Wf-Activity Wf-Step Wf-Task	Workflow Activity	Business Activity Collaboration Activity Business Transaction Activity	Activity	ActionBase	Activity and its specializations	ActionState SubactivityState FinalState PseudoState
Resource	Wf-Resource	Workflow Application	Business Transaction	---	---	Participant	---
Transition	Wf-Transition	Transition Information	Transition Join Fork	ControlLink Join	Sequence While All	All Sequence Foreach	Transition
Object	Wf-Artifact Wf-Data	Relevant Data	Business Document	DataLink Operation Message	Service Operation CorrelationSetDecl	Message	Classifier ClassifierInState
Role	Wf-Role	Workflow Participant	Business PartnerRole	ServiceProvider	---	Participant	---
Actor	Wf-Actor	Workflow Participant	Autorized Role	---	---	Participant	---

Table 1: Alignment of concepts of some process meta-models.

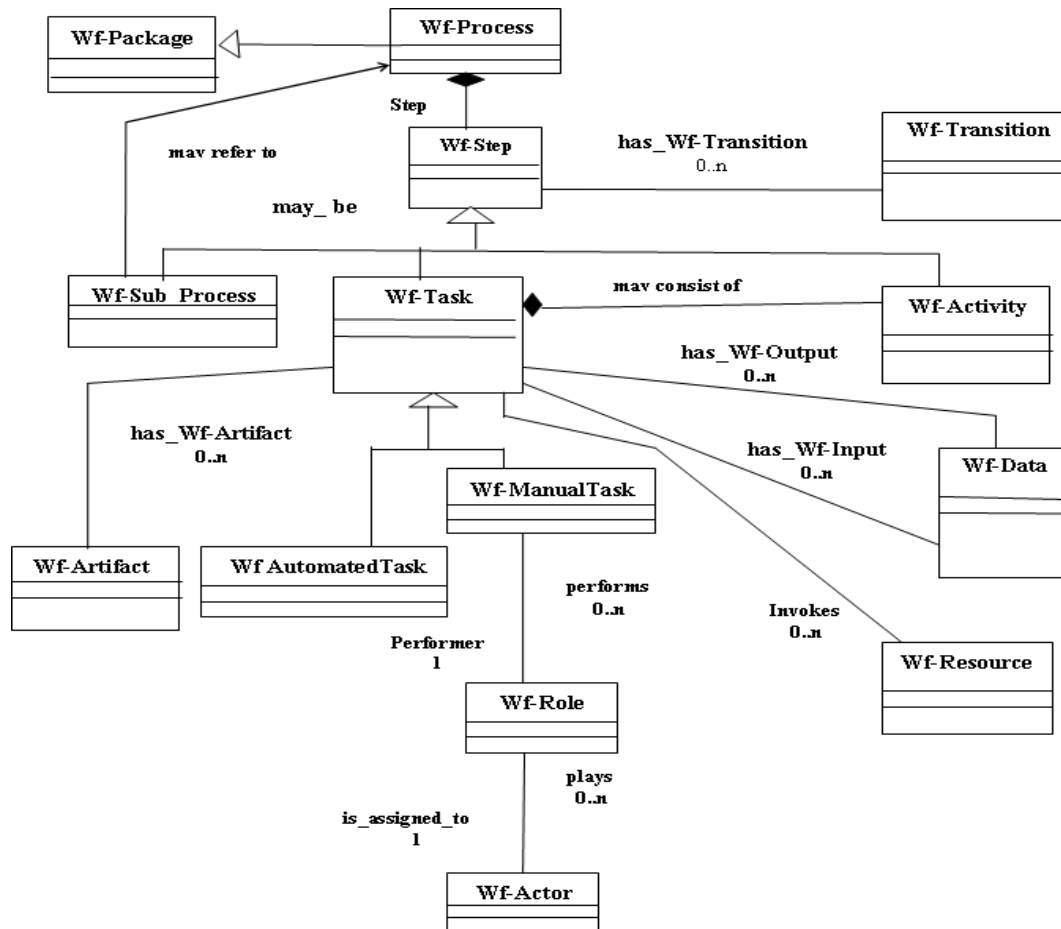


Figure 4: Common Workflow Meta-model.

- ✓ One-to-one mapping: a CWO concept (e.g. CWO:Wf-transition) is referred by an Atomic Concept (e.g. XPDL:TransitionInformation);
- ✓ One-to-many mapping: a CWO concept (e.g. CWO:Wf-transition) can be referred respectively by several concepts (e.g. ebXML: Transition, ebXML: Join and ebXML: Fork) which are enumerated in an Enumerated Concept.

However, these mapping rules may be more complicate: a correspondence between a CWO atomic and a composed concept of one meta-model or between a CWO composed concept and another composed concept of one meta-model. These mapping rules are codified in XML Schemas.

Once the mapping rules are defined for the meta-models, their process models are then be described by the CWO concepts, i.e. the CWO concepts are used as metadata to annotate process semantics. We call the process models described by the CWO metadata as CWO-annotated process models.

4.2 Model Annotation

Based on the meta-models, the CWO concepts will take place the corresponding process modeling constructs to

describe the processes. The models (contents) are instances of meta-models and those instances usually describe certain domains. The representations of domains are often various due to diverse uses of terminology and conceptualization, resulting in semantic heterogeneity of models contents. Domain ontologies are agreed as standard representations and semantic definitions of domain concepts by annotation users. Semantic heterogeneity of models (contents) can be reconciled by referencing ontological concepts represented in domain ontologies. Therefore, we use domain ontologies to annotate the models contents and thesaurus as a type of ontology for annotating the used terms to help the cooperating actors for explicating the meaning of the models.

The annotation method of models is to build relationships between models contents and domain ontologies. The model annotation is therefore to map the concepts defined in the models to those defined in the domain specific ontology. For this, different mapping methods can be used.

▪ Mapping rules in model annotation

Different mapping strategies can be used between concepts in the models and the domain specific ontology.

They can be simple rules applied in meta-model annotation by referring specific model content in modeling constructs to corresponding domain concepts. More complicated mappings can be defined through refined relationships between concepts used in models and concepts defined in domain ontology. The mapping rules are applied using two relationships of mapping:

- ✓ **Simple relationships:** It is a simple mapping by reference. The relationship of mapping can be defined as one type 'refers to'. It assumes that almost all concepts in the model have equal or approximately equal concepts in the ontology. We have adopted such mapping strategy in the meta-model annotation to build the correspondences of concepts between meta-models concepts and the CWO concepts. The strategy of simple reference is easy to apply to map the concepts.
- ✓ **Refined relationships:** The concepts used in process models are variously defined initially for different projects. Therefore, it might be difficult to find equally defined concepts in the domain specific ontology for process models. However, they are still within one domain, there must be some relationships between concepts in models and concepts in ontology. We define some refined relationships to link the concepts between models and ontologies for the model annotation.

The models that are related to domain information are usually artifacts, actors and activities. These concepts will be annotated by domain specific ontology concepts. For this purpose, we use the following relationships for the model annotation.

- The relations: '*kind_of*' and '*phase_of*' are used for annotating activities with domain ontology (i.e. relating an activity with concepts defined in domain ontology).
- The relations: '*member_of*', '*subClass_of*' and '*instance_of*' are used to annotate relationships between roles or actors defined in a model and concepts defined in domain ontology.
- The relations: '*same_as*' and '*equivalent_name*' denote the relationship of synonym. They are used respectively, for linking an activity with concepts defined in domain ontology (concept level) and with thesaurus (terminology level).

5 SCOR as example of reference domain ontology

Checking the literatures of specific domain ontologies in the field of business process, we have found the SCOR model as a process reference model. It provides the process templates and standards of logistics process. It is just referenced as an example of domain ontology

including the domain activities and the domain goals. Its role is to facilitate the model annotation and the goal annotation.

The Supply Chain Operations Reference (SCOR) [36] model is the product of Supply Chain Council (SCC). It is a process reference model that has been developed and endorsed by the SCC as the cross-industry standard diagnostic tool for supply-chain management. It has been therefore developed to describe the standard business activities associated with all phases of satisfying a customer's demand.

There are three level process details in the reference model. The top level defines the scope and content of SCOR. Five process types: Plan, Source, Make, Deliver and Return are defined at this level. The second level is the configuration level defining the core "process categories". The third level is the process element level, decomposing the process categories into the process elements. The process element level consists of process element definitions, process element information inputs, and outputs, process performance metrics, best practices, system capabilities required to support best practices, and systems/tools.

The level 3 process elements provide enough details of references for domain activities. In this paper, we model domain ontology concepts based on the SCOR process elements at level 3 for model annotation purposes. The SCOR ontology is formalized in OWL. To organize those concepts, we categorize domain concepts with the concepts (Wf-Activity, Wf-Artifact, Wf-Role, and Wf-Actor) defined in CWO (Common Workflow Ontology). The purpose of the categories is to establish the mapping relationship between a CPSAM model and the reference ontology when annotation. The concepts are modeled as OWL Classes, and they are organized by a subsumption hierarchy.

For annotating the models, we apply, in our case, "S1 Source Stocked Product" model as reference domain ontology for the item receiving process. Each process element is a task ontology concept, which is referenced by the concept Wf-Activity in the CPSAM. For example, two Workflow models are both about purchase order process domain. In one model, an activity called 'Create Order', while in another model, is called 'Get Order Data'. To enable a same understanding for these activities, we should annotate them by the same concept of the activity SCOR ontology, for instance, 'phase_of Wf-Activity: Schedule Product Deliveries'. That means that are regarded as a phase of the activity ontology 'Schedule Product Deliveries'. Another example, the activities: 'Check Items' and 'Verify Order' are annotated with the activity SCOR ontology 'Verify Product', i.e. 'kind_of Wf-Activity: Verify Product'. For further details on SCOR, we refer the reader to [36].

After the meta-model annotation, the models are then described as a CWO annotated process models.

Therefore, the annotation of models is done directly in the CWO annotated process models instead of original models. Thereby, we can formalize the CWO annotated process models in the common process semantic annotation model (CPSAM).

6 Formalization of a common process semantic annotation model

In this section, we build a common process semantic annotation model (CPSAM), which is based on the CWO concepts to describe a process model. The constructs of CPSAM relate some common constructs often used in most process modeling languages and present the main semantics of process information in the models. We formalize CPSAM as follows:

CPSAM = (AV, AR, AC, AF, AI, AO, AT, RS DO).

Where AV is a set of activities composing a process, AR is a set of roles interacting with a process; AF is a set of artifacts participating in a process. AC is a set of actors (agents, users) participating in the process execution. AI is a set of parameters of inputs of an activity. AO is a set of parameters of outputs of an activity. AT is a set of transition conditions of an activity. RS is a set of resources that are invoked during the execution process. DO is a subset of domain ontology concepts.

An activity is a model fragment in CWO, considered as a step in a process and may be a sub-process or atomic activity. Therefore an annotated activity AV_i is described as follows:

$AV_i = (id, name, same_as, equivalent_name, has_Wf-Role, has_Wf-Actor, has_has_Wf-Artifact, has_has_Wf-Input, has_has_Wf-Output, has_Wf-Transition, has_Wf-Resource, kind_of, phase_of)$

Each element in CPSAM has id and name to uniquely identify the element. 'Same_as', 'kind_of', and 'phase_of', are used to annotate the activities with domain ontology. 'Equivalent_name' provides synonym of the name from terminology level using thesaurus. However, we can add other relationships such as 'is_as', 'step_of', 'subClass_of' and 'instance_of' to relate the concept of Activity with concepts defined in domain ontology.

The relationships: 'has_Wf-Actor', 'has_Wf-Artifact', 'has_Wf-Input', 'has_Wf-Output', denote the relationships between the activity and other related concepts in CWO. The ontology concepts are denoted by URI (Uniform Resource Identifier) in the CPSAM.

- A role is an organizational concept, which may be a supervisor or a manager in an organization that interacts with an activity. The annotated role is represented as follows :

$AR_i = (id, name, same_as, equivalent_name, member_of, subClass_of, instance_of).$

- An actor is a person, agent or user that participates in the process execution. The annotated role is represented as follows.

$AC_i = (id, name, same_as, equivalent_name, member_of, subClass_of, instance_of).$

- An artifact is an object (document, sheet styles, electronic form, etc.), which is manipulated by an activity or a person. It is described as follows :

$AF_i = (id, name, same_as, equivalent_name, related_activity).$

- The inputs and outputs are defined as parameters of an activity, which include data type. They are usually related to artifacts participating in the activity.

$INP_i = (id, name, same_as, equivalent_name, data_type, related_artifact).$

$OUP_i = (id, name, same_as, equivalent_name, data_type, related_artifact).$

- A resource is a physical entity in an organization (tool, machine, etc.) that may be invoked by an activity during the process execution. It is described as follows :

$RS_i = (id, name, same_as, equivalent_name, related_activity).$

- The transition conditions are represented by expressions to constrain the inputs and outputs of the activities. They enable to specify the flux control, which is expressed with logical symbols like AND, OR, and XOR.

$AT_i = (id, name, same_as, equivalent_name, related_activity).$

7 Model annotations for semantic discovery purposes

In a context of inter-enterprise cooperation, the distributed process models should be accessible and reuse in specific projects. For enhancing the interoperability and especially for profiles or goals-oriented queries, we need the consensual representations to specify the semantics of profiles or goals of the models. Therefore, we complement the proposed semantic annotations with profiles and goals annotations.

7.1 Profiles annotation

A profile provides a general description of a process model, intended to be published and shared for facilitating its discovery. Profiles can include both

functional properties (inputs, outputs, preconditions, and post-conditions) and non functional properties (model name, text description, contact information, the location of the model, etc.).

Models profiles concepts are used to annotate intentional usage of process models. In our approach, we build a common profile ontology, which is used to annotate profiles concepts of models. We use the relationship 'related_Wf-Profile' to annotate the models with the common profile ontology. In this way, the common process semantic annotation model (CPSAM) will be extended with this type of annotation and will be formalized as follows:

CPSAM = (AV, AR, AC, AF, AE, AS, AT, RS, DO, PO).

Where PO is a subset of profile ontology concepts and an annotated activity AVi is described as follows:

AVi = (id, name, same_as, equivalent_name, has_Wf-Role, has_Wf-Actor, has_Wf-Artifact, has_Wf-Input, has_Wf-Output, has_Wf-Transition, has_Wf-Resource, kind_of, phase_of, related_Wf-Profile).

7.2 Goals annotation

A goal can be linked to a process model (or a process model fragment). The aim of goal annotation is to facilitate the goal-driven process discovery. The goal annotated models with the global (or common) goal ontology can be queried and reused in specific projects for achieving business objectives (or goals).

Goal annotation of process models is annotating process models with global goal ontology to specify the objectives of processes. Goal ontology is therefore, a set of concepts and relationships between the goals. However, few processes modeling languages and tools support the modeling of the goals as a part of processes modeling, e.g. EEML (Extended Enterprise Modeling Language) [26], which is implemented in Metis tool [27]. Thus, the goals can be modeled and linked to elements of process models. Nevertheless, the representation of the goals and relationships between the goals and the processes are not explicitly represented in many process models.

In the literature of processes models, several goal-oriented modeling approaches [28], [29], [30], [31] were proposed for defining the goal concept. KAOS (Knowledge Acquisition in autOmated Specification) [32] and i*/GRL (Goal-oriented Requirement Language) [33] are considered in the community of the artificial intelligence, and in particular in the requirements engineering as the most significant approaches allowing to model the objectives. The concepts defined in KAOS are: object, action, agent, goal, constraint. Contrary in

i*/GRL, the concepts are the following: actor, role, position and goal.

To express goals, [28] defines verbs key words like those used by KAOS such as: achieve, avoid, maintain. In fact, the use of these verbs leads to a classification of the goals: hard goals and soft software [29], [35]. The hard goals relate to functional goals which are supported by the processes. A functional need defines a potential need that the system must satisfy. The soft goals are related to non functional needs concerning the additional qualities awaited such as performance, quality of service, cost and time.

In a perspective of a goal-oriented cooperation between companies, the creation of a global goal ontology is needed, which provides semantic representations of goals in a consensual way for different organizations. It enables us to build relationships between the local goals and the global goals. Thus, two types of relationships are created to indicate that activity can achieve goals: one relation of type 'achieves_local_but' between the activity and the local goal, and another relation of type 'achieves_global_but' between the activity and the global goal. That means obviously that the process meta-model must preliminary integrate the concept of local goal to facilitate the annotation. Thus, the common process semantic annotation model (CPSAM) will be enriched and extended with this type of annotation and will be formalized as follows:

CPSAM = (AV, AR, AC, AF, AE, AS, AT, RS, DO, PO, GO).

Where GO is a subset of global goal ontology concepts and an annotated activity AVi is described as follows:

AVi = (id, name, same_as, equivalent_name, has_Wf-Role, has_Wf-Actor, has_Wf-Artifact, has_Wf-Input, has_Wf-Output, has_Wf-Transition, has_Wf-Resource, kind_of, phase_of, related_Wf-Profile, achieves_global_but).

However, the goals ontologies belong to the class of the tasks ontologies, which describe tasks or activities for achieving the business objectives. In this context, the tasks ontologies are regarded as goals ontologies. For example, the tasks ontology SCOR contains concepts of goals [36], [20], which are formalized by hard goals and soft goals.

- **SCOR as example of global goal ontology**

The goal ontology in our Workflow domain is also from the SCOR. Usually the hard goals are derived from the level 3 process elements [36] and their inputs and outputs. The performance attributes defined in SCOR are General Soft Goal Category such as reliability, responsiveness, flexibility, cost, and assets. The domain specific soft goals are derived from the metrics of the performance attributes [37]. By analyzing SCOR ("S1

Source Stocked Product” model), several types of hard goals can be derived and extracted from the level 3 process elements [36]. We can quote some examples of goal-oriented activity (called ‘hard goals’): ‘*Sourced Product are On Order*’, ‘*Order is Placed*’, ‘*Order is Validate*’, ‘*Order is Consolidated*’, ‘*Order is Processed*’, ‘*Available Inventory*’, ‘*sourced products are verified*’, ‘*End Items are Delivered*’. Also, there exist goal-roles-oriented such as ‘*Procurement Notification to Supplier*’ and ‘*Payment is Authorized to Supplier*’.

For example, two Workflow models are both about purchase order process domain. The activities for these models: ‘*Check Items*’ and ‘*Verify Order*’, which are annotated as a kind of the activity ontology ‘*Verify Product*’, will be both annotated with the domain goal ontology (i.e, SCOR) by a hard goal ‘*sourced products are verified*’ and also by a soft goal ‘*decrease % defective supplied*’.

Additionally to these main annotation sets for semantic interoperability of Workflow models, namely, meta-model annotation, model annotation, profiles and goals annotation, we introduce another type of annotations.

8 Model aspects annotation

In order to enhance interoperability and to enable a common understanding of the models without any ambiguity; we enrich our semantic annotation framework with complementary annotations that are related to these models. According to [43], [44,] there are different aspects of business process: the informational, the transaction, the security, the quality of services aspect, etc. Which of them are applicable or relevant depends on the context and the purpose of the model. For our concern, we quote the basic aspects: the informational, the functional, the organizational, the behavioural or the resources aspect.

This type of annotation is useful and can then serve for the interoperating actors to know, for example:

- 1) In which the business process is executed, its organizational units and which role or a person of appropriate skills must perform one particular business activity?.
- 2) The order of the execution in which the corresponding activities must be executed.
- 3) What information involved in a business process and how artifacts are propagated among activities?.
- 4) Which resources are required (external applications, computer tool, etc.) for the execution of activities?.

In our work, the proposed annotations refer to several common ontologies. Each one is derived from a common meta-model corresponding to one aspect. Indeed, as argued in [38], the notion of meta-model is strongly

related to the notion of ontology. Thus, these common meta-models are considered as common ontologies and implemented in OWL.

8.1 Informational annotation

The information about models is usually represented in the artifacts manipulating by the activities and containing data, the inputs and outputs as parameters of activities. To annotate the models information, we refer to a common informational ontology, which include the concept ‘Wf-Artifact’ and the concepts ‘Wf-Input’ and ‘Wf-Output’. As we have argued before, this ontology is derived from a common informational meta-model, which is shown in figure 5.

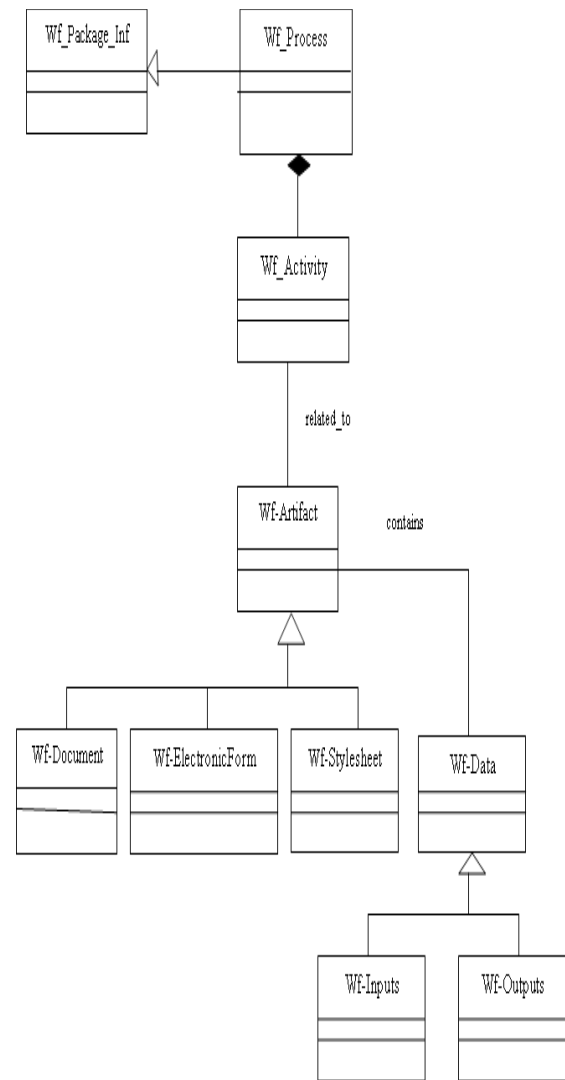


Figure 5: Common Informational Meta-model.

To annotate the activities with a common informational ontology, we use the relationship ‘*related_Wf-Artifact*’. Thus, an annotated artifact and an annotated activity AVi are described as follows:

Wf-Artifact = (*id*, *name*, *equivalent_name*, *same_as*, *related_Wf-Input*, *related_Wf-Output*).

$AV_i = (id, name, same_as, equivalent_name, kind_of, phase_of, related_Wf-Artifact)$.

8.2 Functional annotation

The purpose of the functional annotation is to make explicit the meaning of the functional aspect of a process model. This type of annotation is about the operations (so-called functions) of a process.

Back to the purchase order process domain, when annotating the purchase order process, the annotator refers to an UML activity diagram (figure 6) that contains all the activities of a process. For our case, these activities represent the following operations : 'CreateOrder', 'CheckOrder', 'CancelOrder', 'NotifyCustomer', 'FillOrder', 'ShipOrder', 'CloseOrder', which enable to specify the functional aspect of a purchase order process.

To annotate the functional aspect, we refer to a common functional ontology, which is derived from a common functional meta-model (figure 7). It includes the various functions associated to the activities of a process model. These functions usually correspond to operations carried out by a process. In our approach, a function correspond to an operation (or verb representing the operation) and

an object (for instance, an order) contributing to the operation execution.

Therefore, to link the functions of activities (or operations) to a common functional ontology, we use of the relationship 'related_Wf-function'. Thus, an annotated function and an annotated activity AV_i are described as follows:

$Wf-Function = (id, name, equivalent_name, same_as)$

$AV_i = (id, name, same_as, equivalent_name, kind_of, phase_of, related_Wf-Function)$.

8.3 Behavioural annotation

The behavioural annotation concerns the dynamic aspect of a process. The purpose of such type annotation is, for example, to ensure that cooperating processes have the same behavior (then one can be used instead of the other). In our approach, this type of annotation is provided under the form of enumeration of various passed statecharts by the activities of the process and under form of flow control. To express the common behavior of process models, two common meta-models are proposed:

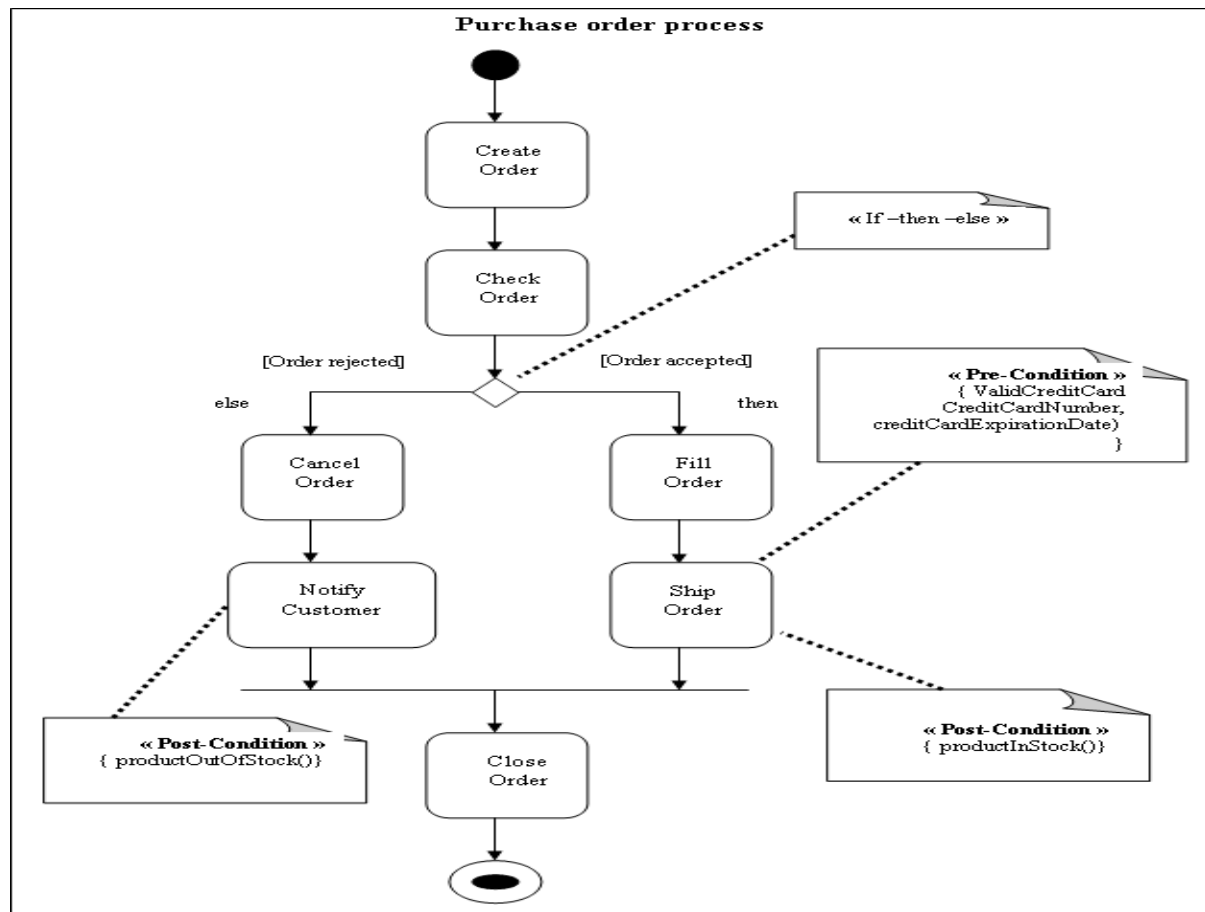


Figure 6: UML Activity Diagram of a Purchase Order Process.

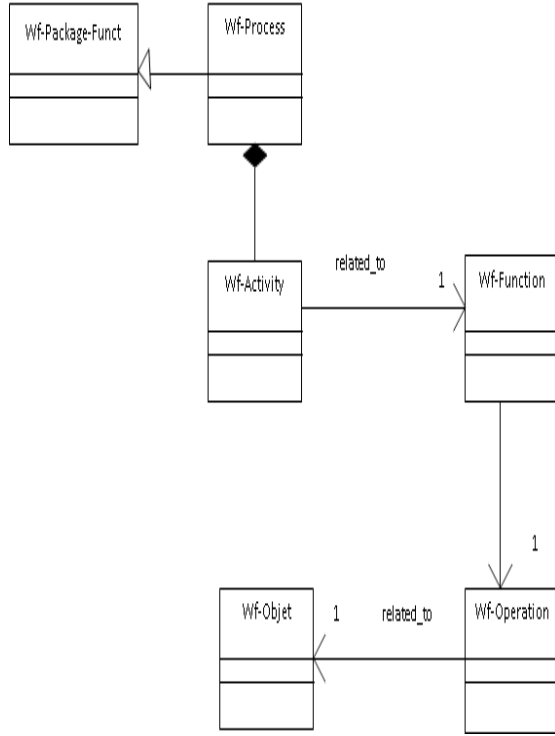


Figure 7: Common Functional Meta-model.

- i) The first one (figure 8) serves to specify the flow control. It enables in particular, to define the flow control, which is usually expressed under form of sequence, parallel and choice.
- ii) The second meta-model (figure 9) is inspired by the statecharts diagram of UML meta-model whose the basic concepts are: StateMachine, State, Transition, Vent, Guard and whose constraints are expressed and formalized by language OCL (Object Language Constraint) associated with UML.

To annotate the dynamic aspect of process models, we refer to both common behavioural ontologies, which are derived from common behavioural meta-models and presented above. The control flow represents the type of the ordering of activities, which is usually described by six operators AND Join, AND Split, XOR Join, XOR Split, OR Join, OR Split. It is described as follows:

$CF_i = (id, name, equivalent_name, related_Wf_Activity).$

Three types of flow control may be distinguished under form of sequence, parallel and choice. For an activity AV_i , we can describe these three flow control as follows:

$Sequence_i = (id, name, equivalent_name, has_inActivity, has_outActivity).$

$Choice_i = (id, name, equivalent_name, has_inActivity, has_outActivity, has_logicConnecteur).$

$Choice_i$ corresponds to the junction or the disjunction at inputs of the activities ($Join_i$) and the element ' $has_logicConnecteur$ ' of $Choice_i$ has the value ' XOR ' for $XOR Join_i$, the value ' OR ' for ' $OR Join_i$ ' and the value ' AND ' for $AND Join_i$.

$parallel_i = (id, name, equivalent_name, has_inActivity, has_outActivity, has_logicConnecteur).$

$parallel_i$ corresponds to the junction or the disjunction at outputs of the activities ($Split_i$) and the element ' $has_logicConnecteur$ ' of $parallel_i$ has the value ' XOR ' for $XOR Split_i$, the value ' OR ' for $OR Split_i$ and the value ' AND ' for $AND Split_i$.

For a certain activity (AV_i), the inputs (Inp_i) and the outputs (Oup_i) are defined as parameters of activity, which include data type. They are usually associated with artifacts manipulated by the activity. We can describe them as follows:

$Inp_i = (id, name, equivalent_name, data-type, related_Wf_artifact).$

$Oup_i = (id, name, equivalent_name, data-type, related_Wf_artifact).$

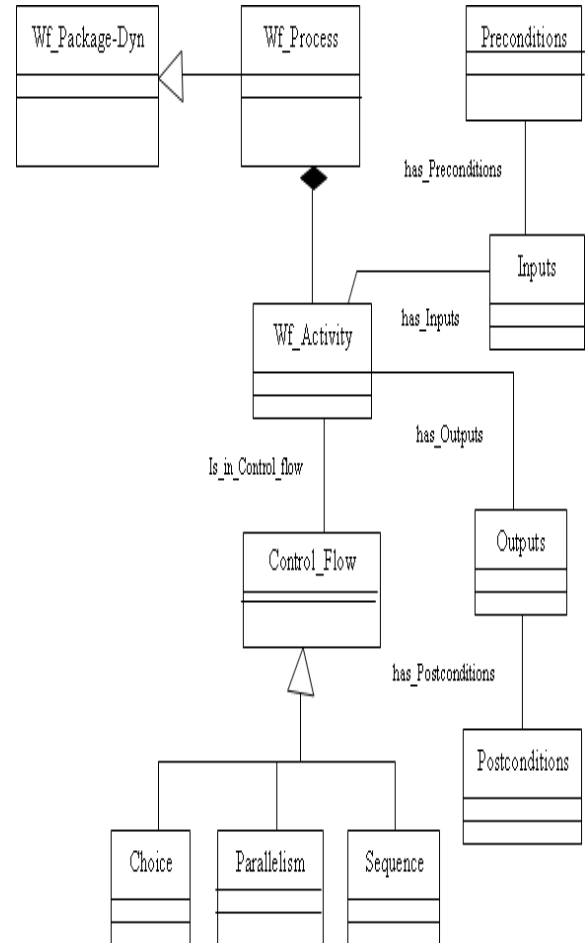


Figure 8: Common Behavioural Meta-model (flow control).

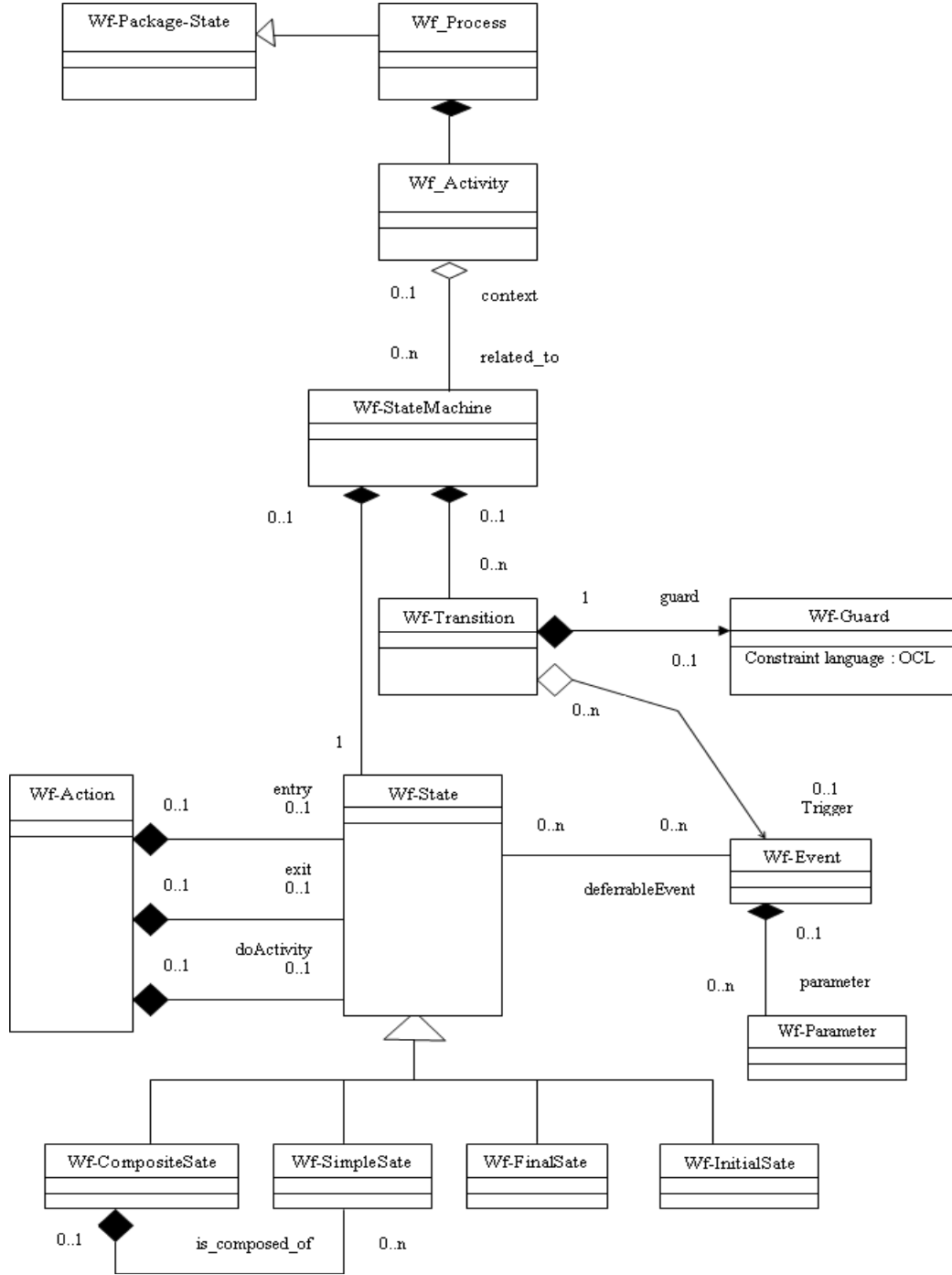


Figure 9: Common Behavioural Meta-model (Statecharts).

In the same way, the preconditions (Pre_i) and the post-conditions ($Post_i$) of the common behavioural meta-model including the flow control can be described as follows :

$$Pre_i = (id, name, equivalent_name, related_Wf_input).$$

$$Post_i = (id, name, equivalent_name, related_Wf_output).$$

Also, the dynamic aspect of process models is expressed through the various states/transitions which we have expressed by the common meta-model of statecharts. Thus, an activity is annotated by referring to its state of transition via the relation "related_Wf-StateMachine". Therefore, an annotated activity AV_i is described as follows:

$$AV_i = (id, name, equivalent_name, same_as, kind_of, phase_of, related_Wf_StateMachine).$$

8.4 Organizational annotation

To annotate the organizational aspect of models, we refer to a common organizational ontology, which is derived from common organizational meta-model (figure 10) using the relationship ‘*related_Wf-OrganizationalUnit*’. This meta-model includes organizational concepts such as organizational unit (department, service, etc.), role, and actor. Thus, an annotated OrganizationalUnit and an annotated activity AVi are described as follows:

Wf-OrganizationalUnit = (id, name, equivalent_name, same_as, is_composed_of).

AVi = (id, name, equivalent_name, same_as, kind-of, phase_of, related_Wf-OrganizationalUnit).

The relationship ‘*is_composed_of*’ refers to organizational concepts defined in common organizational ontology. The annotation of an organizational concept such role, for example, is described as follows:

Wf_Role = (id, name, equivalent_name, same_as, member_of, instance_of)

8.5 Resources annotation

This type of annotation relates to the resources, which are invoked by an activity or a process for its execution. These resources can be material (editor, etc.), software (programs, external applications, etc.), human (well defined qualified persons) or temporal (time envisaged and carried out by the activity). The annotation of resources refers to a common resources ontology, which is derived from a common resources meta-model (figure 11).

We use the relationship ‘*related_Wf-Resource*’ for linking the resources of activities to this ontology”. Thus, an annotated resource and annotated activity AVi are, therefore, described as follows:

Wf-Resource = (id, name, equivalent_name, same_as).

AVi = (id, name, equivalent_name, same_as, kind-of, phase_of, related_Wf-Resource).

To annotate an activity with reference to several common ontologies, we extend our CPSAM by adding a set of common concepts related to models aspects. Thus, the CPSAM will be formalized as follows:

CPSAM = (AV, AR, AC, AF, AE, AS, AT, RS, DO, PO, GO, IO, FO, BO1, BO2, OO, RO).

Where DO is a subset of Domain Ontology concepts, PO is a subset of Profiles Ontology concepts, GO is a subset of global Goal Ontology concepts, IO is a subset of Informational Ontology concepts, FO is a subset of Functional Ontology concepts, BO1 is a subset of Behavioural Ontology1 (related to flow control)

concepts, BO2 is a subset of Behavioural Ontology2 (related to statesharts) concepts, OO is a subset of Organizational Ontology concepts and RO is a subset of Resources Ontology concepts.

Now, we can say that for annotating an activity (*AVi*) of a process model, we use different relationships to link this activity to common ontologies (i.e. relating the activity with concepts defined in domain ontology, goal ontology, informational ontology, etc.).

Therefore, an annotated activity *AVi* is described as follows:

AVi = (id, name, same_as, equivalent_name, has_Wf-Role, has_Wf_Actor, has_Wf-Artifact, has_Wf_Inputs, has_Wf_Outputs, has_Wf_Preconditions, has_Wf_Postconditions, is_in_Control_Flow, has_Wf-State, has_Wf-Resource, kind_of, phase_of, related-Wf_Profile, achieves_global_but).

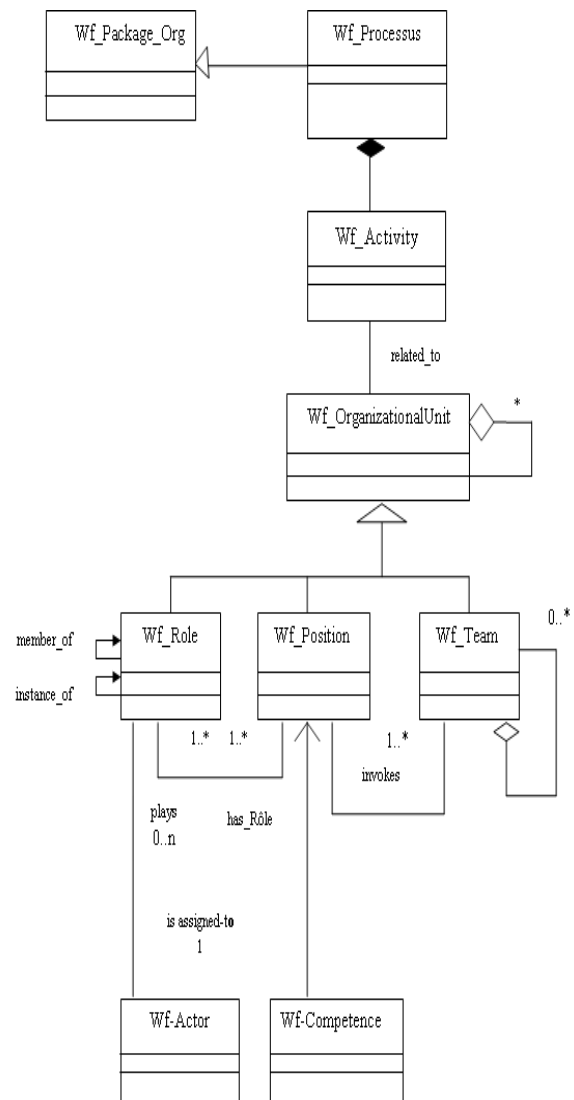


Figure 10: Common Organizational Meta-model.

The relationships: ('*has_Wf-Role*' et '*has_Wf-Actor*') refer to an organizational ontology. The relationships: ('*has_Wf-Artifact*', '*has_Wf-Inputs*', '*has_Wf-Outputs*',) refer to an informational ontology. The relationships: ('*has_Wf-Preconditions*', '*has_Wf-Postconditions*', '*is_in_Control_Flow*', '*has_Wf-State*') refer to a behavioural ontology1 (flow control) and a behavioural ontology2 (statescharts) and the relationship: '*has_Wf-Resource*' refers to resources ontology.

As for the relationships: ('*kind_of*', '*phase_of*'), they enable us to annotate the activity with the SCOR domain ontology and the relationships: ('*related-Wf_Profile*' and '*achieves_global_but*'), they are used to link the activity to the profile ontology and the goal ontology, which is also SCOR in our case.

Based on the above formalization and from a technical view point, these annotations are codified in a XML language with namespace CWO (Common Workflow Ontology), and they are sent by the actor *Act1* to the actor *Act2* for interpretation of the activity.

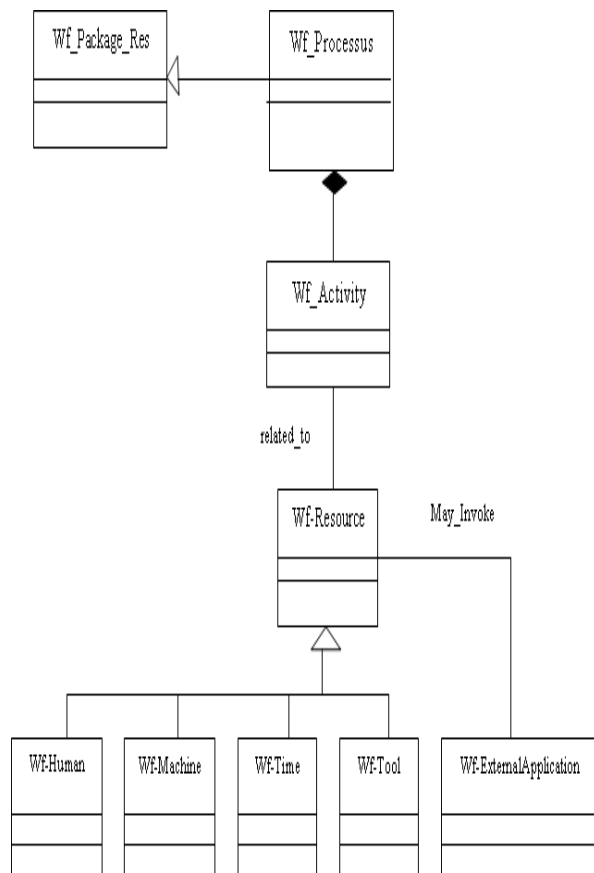


Figure 11: Common Resources Meta-model.

9 Conclusion and Future Work

The interoperability problem is very complex. It is still more complex in the Workflow domain (or process models), which overlaps several fields of application such as the software engineering, the enterprise modeling, the Workflows systems and the Web services

composition. This complexity is mainly due to several aspects, which are by nature related to any Workflow process. Among these aspects, we quote those which are considered basic aspects and concern: the informational, the functional, the behavioural, the organizational or the resources type.

However, the problem of the semantic interoperability of distributed process models can occur in a homogeneous and in a heterogeneous context. To manage this problem on the conceptual level, we have proposed an agnostic approach based on the use of the semantic annotations techniques.

In a homogeneous context, we have used two types of annotations : 1) *structural annotations* : by using only the notation UML, and referring the described concepts in either same ontologies where there is really a consensus between all the partners for all shared concepts, or in ontologies of the partners if there are not completely identical. 2) *Lexical annotations*: by referring to a lexical database such as WordNet, or a WfMC's (Workflow Management Coalition) glossary that contains all the basic vocabulary used in the field of Workflow.

However, to manage the semantic heterogeneity of models within a heterogeneous context, we have proposed a semantic annotation framework allowing the interoperating actors to semantically annotate their Workflow models at different levels: 1) meta-models 2) models content 3) profiles 4) goals 5) models aspects such as the organizational or the informational type.

Among the advantages that our approach offers, we note some of them. First, by externalizing the semantic domain models, we take an agnostic approach to ontology representation languages, as well as of any Web technology. This allows the cooperating actors to annotate their models (or models fragments) according to their preferred ontology representation language. Secondly, by keeping the semantic annotation mechanism separate from the representation of the semantic descriptions, the approach offers flexibility and openness to the developers' community to select their favorite semantic annotation language such as WSDL-S or OWL-S. In addition, this approach enables the traceability of models thanks to their annotations by the successive transformations that may be applied to some initial models.

Obviously, much work in practice for testing the feasibility of the approach and its applicability through applications in business process models integrations in the real world.

For our concern, we have just presented in this paper a theoretical solution for a semantics interoperability problem of process models in a homogeneous and in a heterogeneous context. Certainly, to disclose the technical possibility of the approach, we envisage in

future paper to provide a selected case study in an industrial enterprise.

Therefore, in our future investigations, we are planning to continue our work. First, we implement the different common ontologies in OWL. Second, we create a semantic annotation tool according to our framework, which is currently under development. Then, we apply our semantic annotation procedure as described in section 3.2.

Finally, to support the model annotation, we think that a platform should be developed including, in an integrated manner, facilities or services for the ontology management, the annotation management, etc. (like querying, match-making or browsing an ontology), which where be coupled with annotations services.

References

- [1] S. Hamri, N. Boudjlida and M. Boufaïda (2007). "An approach for building an OWL Ontology for Workflow Interoperability". In Proceedings of the 3rd International Conference on Interoperability of Enterprise Software and Applications, Madera, Portugal, 26-30 March 2007, p. 357—364. R.J. Gonçalves, J.P. Müller, K. Mertins and M. Zelm eds, Springer-Verlag, ISBN: 978-1-84628-857-9.
- [2] S.Hamri (2009). "Bridging MDA and OWL for Workflow Interoperability", In International Review on Computers and Software (IRECOS), ISSN 1828-6003. Vol.4, No. 3, pp 374-381.
- [3] "ebXML (2001), "Business Process Specification Schema Version 1.01", ebXML Business Process Project, <http://www.ebXML.org/specs/ebBPSS.pdf>.
- [4] D.A Bensaber and M. Malki (2012), "Model Driven Approach for Specifying WSMO Ontology", In Proceedings of the 4th Int. conference on Web and Information Technologies, pp 203-213, Algeria, ,
- [5] F.Leyman (2001). "Web Services Flow Language ", IBM Software Group specification, <http://www306.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>.
- [6] OMG (2003), Model Driven Architecture, MDA Guide version 1.0.1, <http://www.omg.org/mda>.
- [7] BPEL4WS (2005). Business Process Execution Language for Web Services version 1.1". <http://www.128.ibm.com/developerworks/library/specification/ws-bpel/>,
- [8] Workflow Management Coalition (2001). Workflow Process Definition Interface, XML Process Definition Language, Document Number TC-1025.
- [9] WfMC (2000). Workflow standard-interoperability, wf-xml binding. Wfmc-tc-1023, v. 1.0.
- [10] OWL-S (2004): OWL Services Coalition OWL-S: Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.1>.
- [11] WSMO(2005), <http://www.wsmo.org/TR/d16/d16.1/v0.21>.
- [12] WSDL-S (2005), <http://lsdis.cs.uga.edu/projects/WSDL-S/wSDL-s.pdf>.
- [13] J. César et al (2006). " Modeling Semantic Web Services: A Case Study", ICWE'06, ACM Palo Alto, USA, pp.32-39.
- [14] J.T.E. Timm, G. C. Gannod (2008). "Grounding and Execution of OWL-S Based Semantic Web Services", IEEE International Conference on Services Computing, Vol. 2, pp.588-592.
- [15] J.T.E.Timm and G.C. Gannod (2005). 'A model-driven Approach for Specifying Semantic Web Services', Proceedings of the 3rd International Conference on Web Services, pp. 313-320,
- [16] G.C. Gannod, et al (2006). "Facilitating the Specification of Semantic Web Services Using Model-Driven Development " in International Journal of Web Services, Vol 3, No 3, pp.61-81.
- [17] Y. Lin (2008). " Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability". PhD thesis, Norwegian University of Science and Technology.
- [18] Y. Lin, H. Ding (2005)." Ontology-based Semantic Web Annotation for Semantic Interoperability of Process Models". In Proc. of the Int'l Conf. on Computational Intelligence for Modeling, Control and Automation, Vienna, Austria.
- [19] Y. Lin et al (2006). "Semantic Annotation Framework to Manage Semantic Heterogeneity of Process Models". In: Dubois, E., Pohl, K. (eds.) CAiSE. LNCS, vol. 4001, pp. 433–446.
- [20] Y. Lin and A.Sølvberg (2007)." Goal annotation of process models for semantic enrichment of process knowledge". In Proc. of the 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007), pp 355–369, Norway. LNCS 4495.
- [21] G.Miller et al (1993), "Introduction to WordNet: An on-line lexical database", <http://wordnetweb.princeton.edu/perl/webwn>.
- [22] <http://wikipédia.org>.
- [23] Workflow Management Coalition (1996), Terminology and Glossary, WfMC Document Number TC-1011, issue 2.0.
- [24] WfMC, <http://www.wfmc.org> (1996).
- [25] Protégé, <http://protégé.stanford.edu/plugin>.
- [26] J.Krogstie,, D.H Jørgensen (2004)." Interactive Models for Supporting Networked Organisations, In Proc. 16th Int. Conf. on Advanced Information Systems Engineering, LNCS 3084, Springer-Verlag 550-562.
- [27] <http://www.troux.com> (2006).
- [28] A.V. Lamsweerde, (2001) "Goal-oriented requirements engineering: a guided tour". In 5th International Conference on Requirements Engineering, pages 249–262.
- [29] N.Mellal (2007). "Réalisation de l'interopérabilité sémantique des systèmes, basée sur les ontologies et les flux d'information", Thèse de Doctorat, Université de Savoie (France).

- [30] Yu, E. J. Mylopoulos (1998). "Why goal-oriented requirements engineering", In: Proc. of the 4th of International Workshop on Requirements Engineering: Foundations of Software Quality <http://www.cs.toronto.edu/pub/eric/REFSQ98.html>.
- [31] M.Lin, H.Guo, J.Yin (2005). "Goal description language for semantic web service automatic composition", In: Proc. of IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2005), Trento, Italy. pp. 190–196.
- [32] R.Darimont et al. (1997). "Grail/kaos: An environment for goal-driven requirements analysis, integration and layout".
- [33] i*: An agent-oriented modelling framework, (2008). <http://www.cs.toronto.edu/km/istar/>
- [34] C. Diamantini and N. Boudjlida (2006). "About Semantic Enrichment of Strategic Models as Part of Enterprise Models. " In J. Eder and S. Dustdar Eds, Proceedings of the 2nd International Workshop on Enterprise and Networked Enterprises Interoperability, in conjunction with the 4th International Conference on Business Process Management. Pages 348--359. LNCS# 4103. Vienna, Austria.
- [35] J. Mylopoulos, L. E.Yu. Chung (1999). "From object-oriented to goal-oriented requirements analysis. Commun", ACM 42, 31–37
- [36] SCOR (2006): SCOR model, version 7, the Supply Chain Council. <http://www.supply-chain.org/page.wv?section=SCOR+Model&name=SCOR+Model>.
- [37] P.Soffer, Y.Wand (2005). "On the notion of soft-goals in business process modeling", Business Process Management Journal 11, pp 663–679.
- [38] J. Bézivin (1998). "Who's Afraid of Ontologies", OOPSLA'98 Workshop, Model Engineering, Methods and tools, Integration with CDIF", Vancouver, October, http://www.metamodel.com/oopsla_cdif-workshop/bezivin1/
- [39] N. Boudjlida and H. Panetto (2007). "Enterprise Semantic Modelling for Interoperability", In Proceedings of the 12th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2007, September 25-28, Patras, Greece.
- [40] M. Elias, et al (2010). "A Business Process Metadata Model for a Process Model Repository. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.), Lecture Notes in Business Information Processing, vol. 50, pp. 287–300, Springer.
- [41] M.Elias, P.Johannesson (2012). "An Empirical Assessment of the Effect of Context-Based Semantic Annotation on Process Model Discovery". CAiSE Workshops 2012: 366-382, Gdańsk, Poland, June 25-26, ISBN 978-3-642-31068-3, Springer,.
- [42] M.Elias, P.Johannesson (2012). "A survey of process model reuse repositories". In: Dua, S., Gangopadhyay, A., Thulasiraman, P., Straccia, U., Shepherd, M., Stein, B. (eds.) ICISTM, Communications in Computer and Information Science, vol. 285, pp. 64–76, Springer.
- [43] B. Axenath, E. Kindler and V. Rubin (2005). "The Aspects of Business Processes: An Open and Formalism Independent Ontology", Technical report, Department of the University of Paderborn, Germany, <http://www.upb.de/cs/kindler/publications/copies/AKR05.pdf>
- [44] J.Gray and A.Reuter (1993). "Transaction Processing : Concepts and Techniques ".
- [45] M. Dimitrov, Alex Simov, Sebastian Stein, and Mihail Konstantinov (2007). "A BPMO Based Semantic Business Process Modelling Environment". In Workshop on Semantic Business Process and Product Lifecycle Management, volume 251 of CEUR Workshop Proceedings. CEUR-WS.org,
- [46] M. Born, , et al (2009). "Supporting Execution-level Business Process Modeling with Semantic Technologies. " In Proc. of the 14th DASFAA Conference. LNCS 5463, Springer.
- [47] C. Di Francescomarino, Chiara Ghidini, Marco Rospocher, Luciano Serafini, and Paolo Tonella (2008). "Reasoning on semantically annotated processes". In Proceedings of the 6th International Conference on Service-Oriented Computing, vol 5364 of Lecture Notes in Computer Science, pages 132–146.
- [48] D. Calvanese, , et a (2012). "Ontology-Based Governance of Data-Aware Processes". In Proc. of the 6th Int. Conf. on Web Reasoning and Rule Systems, LNCS 7497, Springer.
- [49] F. Smith, Proietti, M. (2013). "Rule-based Behavioral Reasoning on Semantic Business Processes". In Proc. of the 5th Int. Conf. on Agents and Artificial Intelligence, SciTePress.