An Ontology-Based and Model-Driven Approach for Designing IT Service Management Systems

María-Cruz Valiente, University of Alcalá, Spain

Cristina Vicente-Chicote, Technical University of Cartagena, Spain

Daniel Rodríguez, University of Alcalá, Spain

ABSTRACT

Currently, few projects applying a Model-Driven Engineering (MDE) approach start from high-level requirements models defined exclusively in terms of domain knowledge and business logic. Ontology Engineering (OE) aims to formalize and make explicit the knowledge related to a particular domain. In this vein, this paper presents a modeling approach, formalized in ontological terms, for defining high-level requirements models of software systems that provide support for the implementation of Information Technology Service Management Systems (ITSMSs). This approach allows for: (1) formalizing the knowledge associated to the ITSM processes contained in an ITSMS; (2) modeling the semantics of the activities associated to these processes in terms of workflows; (3) automatically generating the high-level requirements models of the workflow-based software systems needed to support (part of) the ITSM processes; and (4) from the latter, obtaining lower-level models (and eventually code) by means of automated model transformations. A real case study describing the use of this proposal to model an Incident Management System is also included to demonstrate the feasibility and the benefits of the proposed approach.

Keywords: *Business Management, Incident Management Systems, IT Service Management, Knowledge Management, Model-Driven Engineering*

1. INTRODUCTION

In this day and age no one can question the importance of *Information Technology* (IT) in the business world. A world in which there is a market that is more and more competitive. With the continuous integration and standardization of new computer technologies, the business world is changing frequently. Therefore the business world is immersed in a cycle of continuous improvement, where essentially, the level of quality of the IT services delivered to the customers is often the deciding and differentiating factor. Thus, business people

DOI: 10.4018/jssmet.2011040104

have increased their expectations related to the IT department, and now they need IT to support their business processes in a strategic way. That is, organizations are aware of the closer relationship and convergence between business and IT.

In this vein, *IT Service Management* (ITSM) provides a set of specialized organizational capabilities and a professional practice, supported by an extensive body of knowledge, experience and skills for providing value to customers in the form of IT services (OGC, 2007).

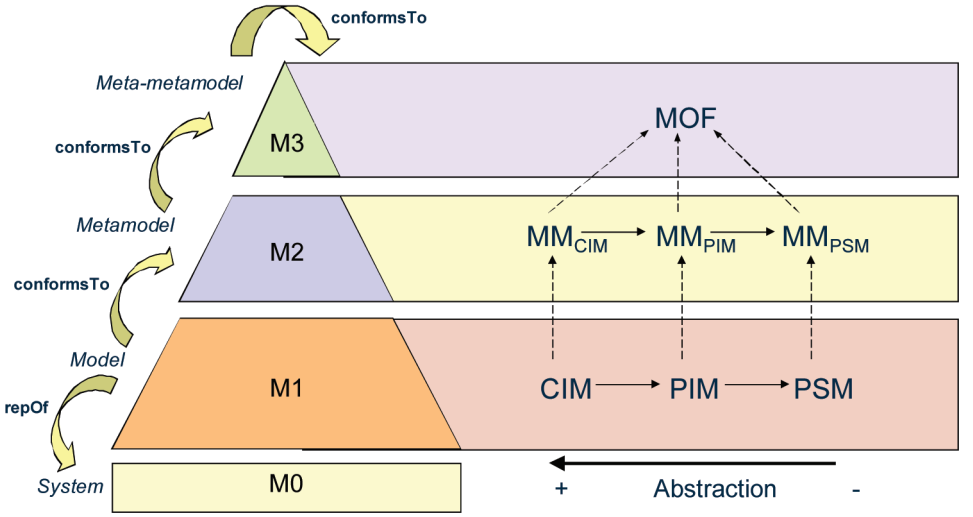
The implementation of any IT service-oriented software system requires performing a number of different steps in order to produce all the required artifacts (either internal or deliverable). Based on the notion that a software system is a representation of another system (i.e., the real-world), the first step is to formalize the domain concepts and the relationships between them (i.e., the ontology), in order to obtain a common vocabulary agreed by all the stakeholders involved in a given project for requirements elicitation. In addition, apart from the domain concepts, additional rules, constraints and semantics are required in order to avoid semantic ambiguities, uncertainties and contradictions. The *Web Ontology Language* (OWL) (Smith, Welty, & McGuinness, 2004), the *de facto* standard for ontology representation, enables the definition of rules, constraints and semantics in terms of logic based domain concepts. The importance of an investigation of the issues involved in the IT service-oriented requirements analysis is also remarked by Lichtenstein, Nguyen & Hunter (2004). However, in spite of the efforts of the *Software Engineering* (SE) community to define new intuitive and powerful techniques, there is still an open gap regarding the automated and seamless integration of domain aspects (i.e., the business view) into the software development process.

The emerging *Model-Driven Engineering* (MDE) paradigm offers a promising solution to cope with this limitation. MDE addresses the inability of third-generation languages to cope with increasing software complexity, allowing us to describe domain concepts effectively (Schmidt, 2006; Gašević & Hatala, 2010). *Model Driven Engineering* (MDE) is a software and system construction approach based on high-level abstract modeling. All the relevant information in a project is stored in models based on well-defined languages and development is then carried out as a sequence of model transformations. The MDE term was first proposed by Kent (2002) but it is derived from the OMG's *Model Driven Architecture* (MDA) initiative (OMG, 2003).

As shown in Figure 1, MDA defines a particular MDE process aimed at separating the business logic from the technological platforms. Thus, organizations can use MDA to meet the integration challenges posed by new platforms, while preserving their investments in existing business logic. MDA proposes three modeling levels, namely (ordered from highest to lowest levels of abstraction): *Computation Independent Model* (CIM), *Platform Independent Model* (PIM), and *Platform Specific Model* (PSM). Different *Model-to-Model* (M2M) transformations among these abstraction levels can be defined either top-down or bottom-up. Commonly, each CIM (model gathering high-level business requirements, sometimes called a domain model) is transformed into one or more PIMs (platform-independent architectural models). Similarly, each PIM is transformed into one or more PSMs (one for each target platform). PSMs are commonly very low level models, enabling the definition of a direct *Model-to-Text* (M2T) transformation for automatically generating the final system implementation, including code, documentation, etc.

At the model layer (M1), CIMs are commonly high-level business models that represent the high-level requirements for the system to build (M0). A high-level requirement is focused on the actual stakeholders problems and needs and describes the characteristics of the domain of the systems (that is, what is needed, but not how this is to be implemented) (Olivé, 2007). Therefore, CIMs help in bridging the gap between the conceptual level mainly performed by domain experts and the implementation level performed by the designers and develop-

Figure 1. MDA and the OMG's four-layers metamodeling pyramid as depicted in (Vicente-Chicote & Alonso, 2007)



ers of the artifacts that together satisfy the domain requirements (OMG, 2003). However, most actual MDA projects are focused on PIM-to-PSM transformations or PSM to code transformations, and they rarely get a comprehensive understanding of the problem domain.

In this paper, we present a formal ontology-based and model-driven approach for the semantic enrichment of workflow models in the context of *Information Technology Service Management Systems* (ITSMSs). This approach aims to enable the creation of enriched machine-processable workflow models, which can be later processed and reused by other workflow modeling tools. In addition, our proposal aims at supporting a set of (automated) model transformations from the ontology-based workflow models to CIM models describing the high-level requirements of the software systems supporting the activities defined as part of the workflows in an ITSM process. An ITSM process is defined as *"a structured set of activities designed to accomplish a specific objective"* (WfMC, 1999).

For this purpose, we use the workflow ontology defined by Valiente, García-Barriocanal, and Sicilia (2011) which allows us to formally describe ITSM processes in terms of workflow models. This ontology uses OWL, combined with the *Semantic Web Rule Language* (SWRL) (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, & Dean, 2004), the latter used to specify additional constraints and infer new knowledge. Then, we transform the resulting OWL workflows into *Business Process Model and Notation* (BPMN) models (OMG, 2010a), which can be further processed and manipulated using the BPMN modeler included as part of the Eclipse *SOA Tools Platform* (STP) project (Eclipse, 2006). To do so, an *XML Metadata Interchange* (XMI) file (OMG, 2007) extracted from an OWL workflow is transformed into a XMI file that represents a BPMN model using an *Extensible Stylesheet Language Transformations* (XSLT) script (Clark, 1999).

The main objective of the research presented in this paper is to translate ITSM process models (expressed using natural language or informal graphical representations) into formal and comprehensive representations of the *Information Technology Infrastructure Library* (ITIL) (http://www.itil-officialsite.com/) processes described in terms of the workflow ontology, so that they can be used as the basis

for modeling and designing the software systems that underpin an ITSMS.

As a proof of concept, we started a pilot project with a Spanish IT service provider (the *Information and Communication Technology Department* – ICTD – of a Spanish company) interested in improving the quality of the services they were delivering to their customers in order to obtain an optimal level of customer satisfaction and to become more competitive and efficient.

The rest of the paper is organized as follows. Section 2 covers the background. Then, Section 3 presents the proposed approach for implementing workflow-based ITSM processes contained in an ITSMS, exemplifying it with a case study regarding an *Incident Management* process based on the ITIL framework. Section 4 reviews related work. Finally, Section 5 draws some conclusions and outlines some future work.

2. BACKGROUND

2.1. IT Service Management Systems

The concept of 'service' is understood differently depending on the domain or application area in which it is used. This sometimes leads to confusion, as explored in (Jones, 2005; Ferrario & Guarino, 2009). The IT Service Management Forum (itSMF) defines an IT service as *"a service provided to one or more customers by an IT service provider. IT services are based on the use of information technology and supports the customer's business processes. IT services are made up from a combination of people, processes and technology and should be defined in a Service Level Agreement (SLA)"* (itSMF International, 2007). Therefore, in this context, IT services can be considered commitments (Ferrario & Guarino, 2009).

An *IT Service Management System* (ITSMS) is a collection of interrelated and coordinated rules, principles and activities, structured in form of processes (Nextel, 2010). According to ISO/IEC 20000 (International Organization for Standardization, 2005a, 2005b), the quality international standard for ITSM, an ITSMS must include *"policies and a framework to enable the effective management and implementation of all IT services"*:

- *Management Responsibility:* Through leadership and actions, IT service providers must prove its commitment to developing, implementing and improving its ITSM capability within the context of the organization's business and customers' needs.
- *Documentation:* IT Service providers must provide documents and records to ensure effective planning, operation and control of ITSM.
- *Competence, awareness and training:* All ITSM roles and responsibilities must be defined and maintained together with the competencies required to execute them effectively. Also, training needs must be reviewed and managed to enable staff to perform their role effectively. Finally, IT service providers must ensure that its employees are aware of the relevance and importance of their activities and how they contribute to the achievement of the ITSM objectives.

There are several well-established good practice frameworks to create an effective ITSM, such as ITIL (in fact, ISO/IEC 20000 is based heavily upon the ITIL framework). Nowadays, ITIL is the best-known and most widely accepted best practices (Hochstein, Zarnekow, & Brenner, 2005) and it has become the *de facto* standard for ITSM. ITIL provides *"a detailed description of a number of important IT practices, with comprehensive checklists, tasks, procedures, and responsibilities, which can be tailored to any IT organization"* (OGC, 2007).

ITIL revolves around processes required to manage IT services, and they are the key to efficiency, effectiveness and the ability to improve

them (OGC, 2007). ITIL process definitions describe tasks, dependencies and sequence that can be modeled in terms of workflow models for IT service-oriented requirements elicitation. As mentioned earlier, IT service-oriented requirements analysis is of major importance in ITSM, but very little attention has been given to investigating the diverse issues involved in it (Lichtenstein, Nguyen, & Hunter, 2004).

A workflow can be defined as *"the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules"* (WfMC, 1999). Workflow models provide a simplified view or description of the business structure and capture the business core functions (Eriksson & Penker, 2000). Workflow models do not necessarily include any detail about the software systems. However, when a software system is designed to automate (part of) the business process, its requirements can be derived from (part of) the corresponding workflow model Eriksson & Penker, 2000). Nowadays, there are several graphical notations that support workflow modeling, such as the *Business Process Model and Notation* (BPMN) (OMG, 2010a), the *Event-driven Process Chain* (EPC) (Sheer, 2000) and *Activity Diagrams* of the *Unified Modeling Language* (UML) (OMG, 2010b). BPMN is currently considered the *de facto* standard notation for business processes modeling, and it is possible to find several workflow management systems described using this notation.

2.2. Ontologies

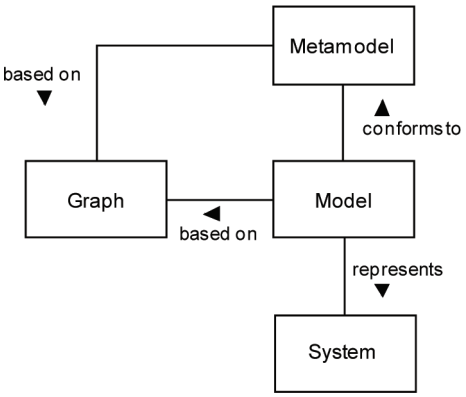
Ontologies (Gruber, 1995; Uschold & Grüninger, 1996) are explicit representations of a shared conceptualization. In this context, the term 'shared' indicates that an ontology captures some consensual knowledge, and the term 'conceptualization' means an abstract, simplified view of a domain of discourse (that is, the real-world) (Gašević, Djurić, & Deved, 2007). There may exist several conceptualizations, and thus ontologies, for the same domain (Olivé, 2007).

Ontologies are sometimes claimed to be the next silver bullet in knowledge modeling, aiming at avoiding conceptual ambiguities, advocating reuse and standardization, and serving as building blocks for more complex automated-reasoning systems (Gruber, 1991; Chandrasekaran, Josephson, & Benjamins, 1999). *Ontology Engineering* (OE) has shown to be useful for (KBSI, 1994): (1) consensus building; (2) object-oriented design and programming; (3) component-based programming; (4) user interface design; (5) enterprise information modeling; (6) business process reengineering; and (7) conceptual schema design.

Since the inception of the Semantic Web, in which ontologies are the principal resource to integrate and deal with online information, a new set of standards have been proposed. The *Web Ontology Language* (OWL) (Smith, Welty, & McGuinness, 2004) is one of such standards that belong to a family of knowledge representation languages prepared for the Semantic Web (although this language can be adopted in other domains, as we propose in this thesis). OWL has reached the status of *World Wide Web Consortium* (W3C) recommendation. From a technical point of view, OWL extends the *Resource Description Framework* (RDF) and RDF Schema (RDF-S), allowing us to integrate a variety of applications using the *Extensible Markup Language* (XML) as interchange syntax. Therefore, due to its RDF basis, OWL ontologies can be associated to any other form of information expressed on the Semantic Web.

A related specification, the *Semantic Web Rule Language* (SWRL) (Horrocks, Patel-Schneider, Boley, Tabet, Grosof, & Dean, 2004), is based on RuleML (n. d.). The SWRL extends the OWL, providing logic-based rules and, in consequence, providing more expressiveness. Rules together with stored facts (knowledge base) are executed as inputs by the rule engine, which infers new facts as an output. In addition, if the rule engine infers new knowledge using

Figure 2. Basic MDE principles



forward chaining, this knowledge can be used for further inference.

Ontology Development Environments

Graphical ontology editors allow us to build formal ontologies. Graphical ontology development environments integrate an ontology editor with other tools and usually support multiple ontology representation languages. They are aimed at providing support for the whole ontology development process and for the subsequent use of the ontology (Corcho, Fernández-López, & Gómez-Pérez, 2002).

The open source Protégé (http://protege.stanford.edu/) tool is an example of a widespread ontology development environment. The Protégé-OWL editor is an extension of Protégé that provides support to OWL. The Protégé-OWL editor enables users to load and save OWL and RDF(S) ontologies, edit and visualize classes, properties, taxonomies and several restrictions, as well as class instances (i.e., the actual data in the knowledge base). It also includes the *SWRLTab* which is an extension for editing and executing SWRL rules in conjunction with the Jess rule engine (http://www.jessrules.com/).

2.3. Matching Ontologies and Conceptual Models with Metamodels

A model in MDE is a *"graph-based structure representing some aspects of a given system and conforming to the definition of another graph called a metamodel"* (Bézivin, 2005). Therefore, the basic set of MDE principles is based on two concepts and two basic relations. The two concepts are *system* and *model* and the relations are *conformance* and *representation*: a model is said to represent the system and a model is said to conform to its metamodel. These principles can be seen in Figure 2.

In the context of MDE, we must be clear about the structure of a domain (that is, the ontology) related to the system to build, so that we can formalize this structure or its relevant part in terms of a metamodel for any attempt at automation in the software development process (Stahl & Völter, 2006). According to MDE, ontologies (that is, the OE part) would cope with the 'repOf' (representation of) relation that exists between models and systems (Figure 1). A metamodel through the abstract syntax defines concepts, attributes and relationships that help a model conform more closely

to the system that it represents. That is, the abstract syntax consist of *"a definition of the concepts, the relationships that exist between concepts and well-formedness rules that state how the concepts may be legally combined"* (Clark, Sammut, & Willans, 2008).

Just like the approach of (Ruiz & Hilera, 2006), we consider that ontologies and metamodels have different purposes: ontologies are descriptive and they belong to the structure of a domain (that is, the real-world), whereas metamodels are prescriptive and they belong to the MDE solution (that is, the application to be performed). However, conceptual modeling of software systems is comparable with ontologies, because they share some modeling principles. A conceptual model captures the semantics for a given application domain, and ontologies are supposed to capture semantics about real-world domains, independently from specific application needs. Similarly to the approach of Bézivin (2009), we also consider a metamodel as a simplified ontology in the sense that it is a set of concepts and relations between these concepts. Therefore, ontologies can act as the basis for defining *Domain Specific Languages* (DSLs) in terms of a metamodel in order to generate conceptual models for the implementation of specific software systems. DSLs are composed of a metamodel, including its static semantics, and a corresponding concrete syntax (Stahl & Völter, 2006), specially designed for the MDE solution. Since a DSL describes domain knowledge it requires detailed knowledge about the real-world domain (that is, the ontology). Just as remarked by Devedžić (2002), if ontologies are not used, different conceptual models of the same domain could be incompatible, even if they use the same DSL for the implementation of the related software systems. Ontologies guarantee that all models follow the same principles and constraints. That is, *'Ontological metamodeling'* has an unambiguous mapping between the universe of discourse and the words and symbols that name and describe it (Goeken & Alter, 2009).

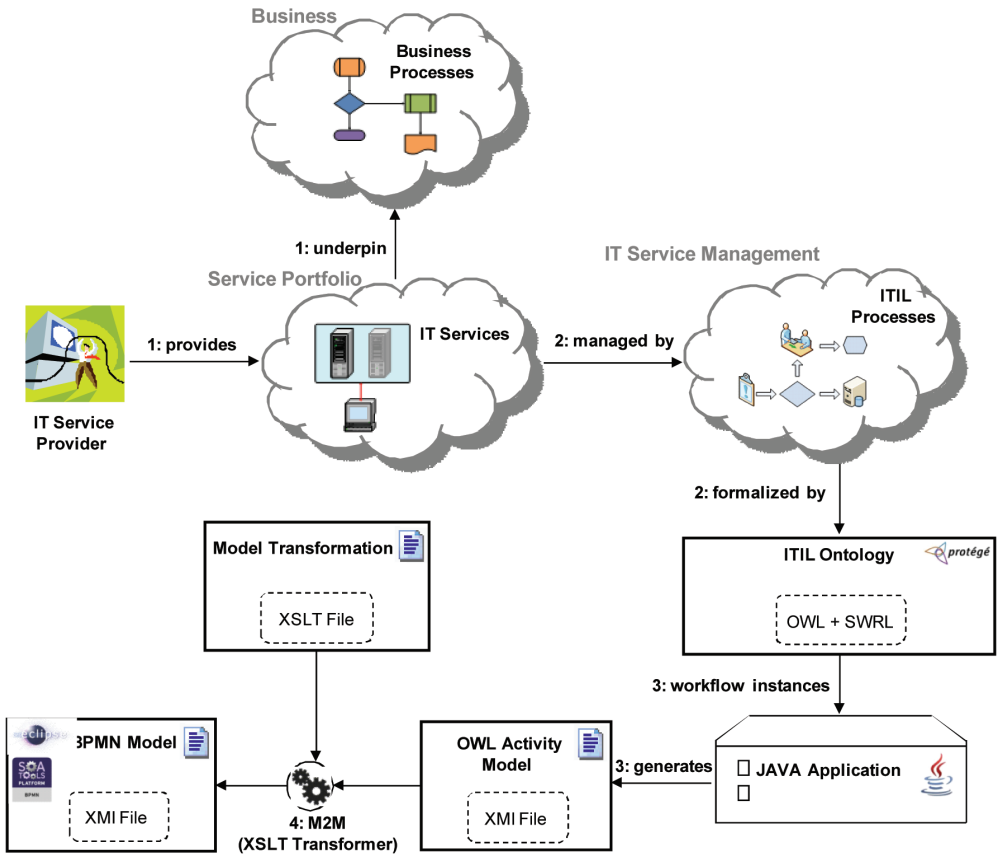
3. AN APPROACH FOR IMPLEMENTING ITSM PROCESSES

As mentioned earlier, IT service requirements analysis is a critical issue in the implementation of software systems that underpin the ITSM processes contained in an ITSMS. In this section, we discuss the approach adopted to integrate the workflow ontology presented in (Valiente, García-Barriocanal, & Sicilia, 2011) with the development of IT service-oriented software systems using a model-driven approach. The importance of using ontologies to automate and validate service process models is also remarked by (Verma & Sheth, 2007). As also stressed by Verma and Sheth, OE can provide *"a basis of building models of all things in which computing is interested"* (Verma & Sheth, 2007). It is also worth noting that a formal description of the functionality associated to a service process is crucial for its reuse (Verma, Sivashanmugam, Sheth, Patil, Oundhakar, & Miller, 2005), while a formal description of the data it exchanges is a key requirement for interoperability (Nagarajan, Verma, Sheth, Miller, & Lathem, 2006). As shown in Figure 3, the proposed approach consists of four steps, which are discussed in the following subsections.

3.1. Service Portfolio

We start with the fact that the IT services are contained within a service portfolio belonging to an IT service provider. These IT services underpin the business processes of different organizations. For example, starting with our pilot project, an instance of the *itil:ITServiceProvider*, *itil:ICTD_provider*, provides several IT services, which are contained within *itil:ICTD_ServiceCatalog*: *itil:Access3G*, *itil:Backup*, *itil:MailingLists*, *itil:DataNetwork*, *itil:Microcomputing*, *itil:SWManagement*, *itil:SWLicensing*, *itil:Staff_email*...

Figure 3. Architecture of the Ontology-based and Model-driven ITSM approach



3.2. ITIL-Compliant and Ontology-Based IT Service Management

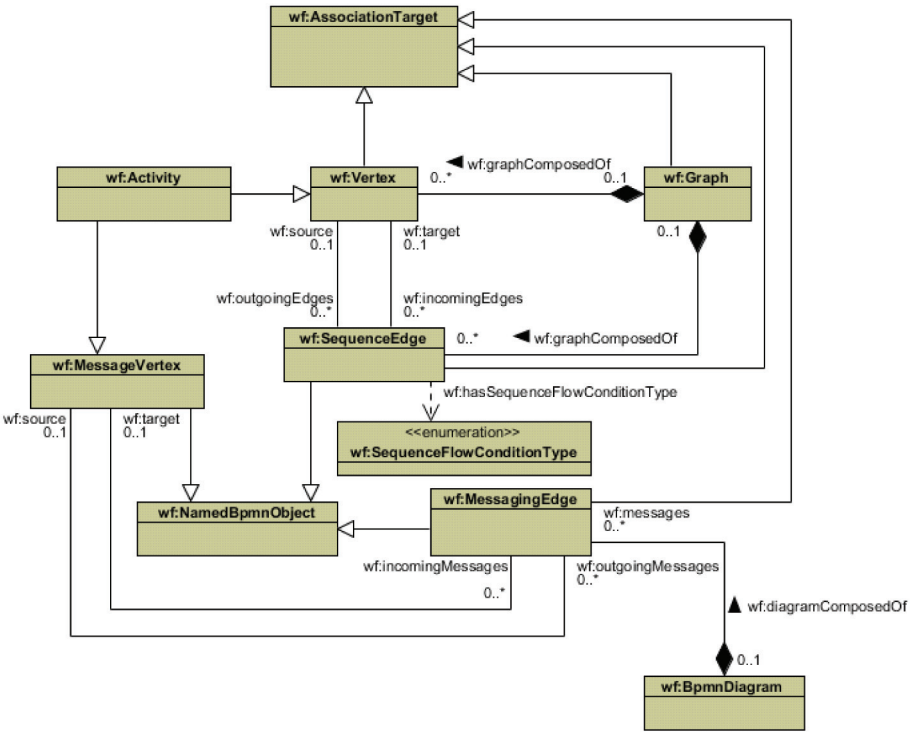
In order to assess the efficiency and quality of the IT services included in the service portfolio, a complete ITSM model (see the M0 layer in Figure 1) is carried out according to the ITIL ontology defined by Valiente, García-Barriocanal, and Sicilia (2011). This model relies on the *ITIL V3 Service Management Model*. It provides mechanisms for semantic analysis (based on the underlying constraints), new knowledge inference, and SLA management, among others.

In our pilot project, the Spanish company decided to start adopting ITIL and to implement the *Incident Management* process (*itil:IncidentManagement*), adapting it according to its business requirements using our approach (the *itil:ICTD_IM_Process* instance of the *itil:IncidentManagement* class). This process was selected to validate our work because the *Incident Management* process is highly visible to the business and, therefore, it is often one of the first processes to be implemented in ITSM projects (OGC, 2007). Also, this process is a relatively simple one with a reasonable number of classes and properties associated.

3.3. Workflow Modeling

In order to provide support to the implementation of the ITIL processes, we use the workflow ontology included as part of the ITIL ontology in (Valiente, García-Barriocanal, & Sicilia, 2011)

Figure 4. UML class diagram representing the workflow ontology



for defining the workflow models associated to each ITIL process (see the M0 layer in Figure 1). The workflow ontology that is shown in Figures 4 and 5 is a formalization in OWL of the BPMN constructs (OMG, 2006a).

The *wf:BpmnDiagram* is the workflow representation (i.e., the workflow model) in form of a BPMN diagram which is composed of pools (*wf:Pool*) and messages (*wf:MessagingEdge*). In our approach, we consider *itil:Activity* a subclass of *wf:BpmnDiagram* in order to model the high level requirements of the software system that could automate the activities defined as part of a workflow model associated with an ITSMS.

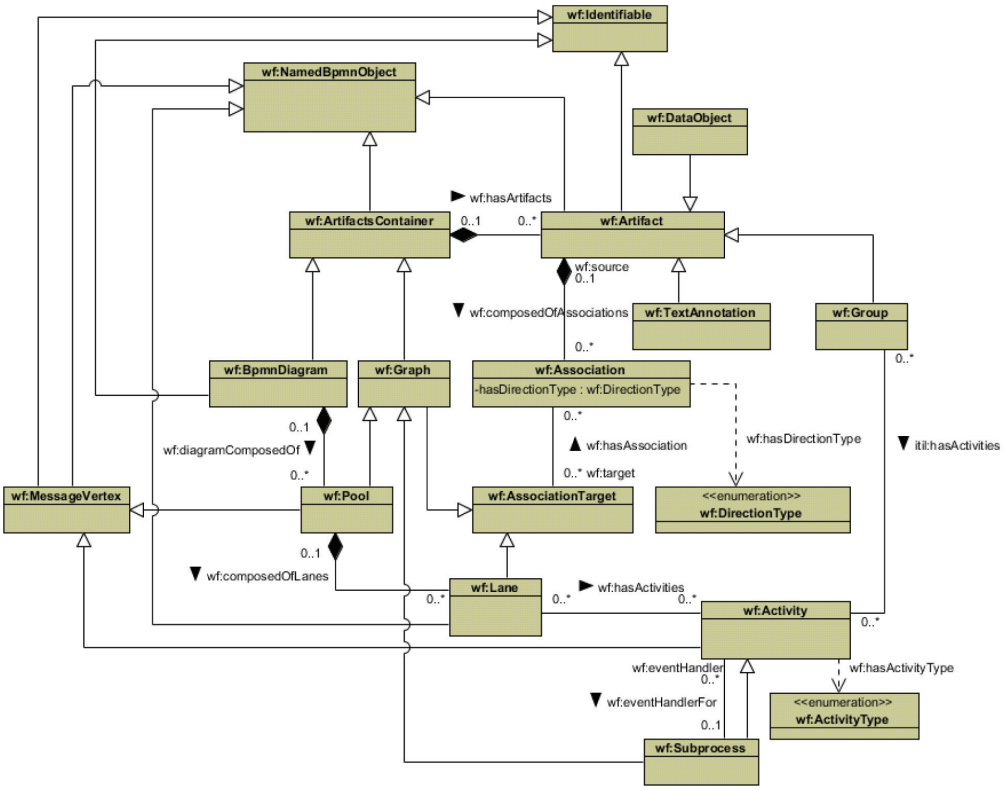
A complete specification of a BPMN diagram definition in the workflow ontology, which we cannot describe in detail for reason of space, consists of the next model elements: *Artifacts* (*wf:DataObject*, *wf:Group* and *wf:TextAnnotation*), *Graphs* (*wf:Pool* and *wf:Subprocess*), *Lanes*, *Nodes* (*wf:Activity*) and *Edges* (*wf:Sequence*Edge and *wf:Messaging*Edge).

For example, in our pilot project, the workflow associated with the *Incident Management* process (*itil:ICTD_IM_Activity*), that is shown in Figure 6, was defined in terms of a BPMN diagram (Figure 7) in the workflow ontology using the Protégé 3.4.4 ontology editor. In this case, we only have one pool instance (*itil:ICTD_Pool_IncidentManagement*) associated with the subprocess instance (*itil:ICTD_IncidentManagementSystem*) that contains all the elements of the workflow (Figure 8).

3.4. Workflow Model Transformation

To manage the knowledge related to the ITIL process that is being automated through computer tools (*itil:Application*) (from here on, we

Figure 5. UML class diagram representing the workflow ontology (cont.)



use the prefixes 'itil' and 'wf' to refer to the namespaces of ITIL and workflow respectively), those activities (*itil:Activity*) defined in the ITIL ontology can be included in the Eclipse platform for its total (or partially) automation by means of a software system. To accomplish this, a Java application is implemented which, (1) shows all of the instances of *itil:Activity* defined in the ontology; (2) allows the user to establish which of these activities will be automated and implemented in the *itil:Application* as part of the ITSMS; and (3) executes an XSLT script to transform the selected activities into a BPMN model (see CIM at the M1 layer in Figure 1), which conforms to the BPMN metamodel (see MM_{CIM} at the M2 layer in Figure 1), obtained from the Eclipse BPMN modeler subproject developed for the *SOA Tools Platform* (STP) project (Eclipse, 2006). The resulting BPMN model describes, at a very high-level of abstraction, the IT service-oriented requirements to be implemented as part of the ITSMS. Since the BPMN modeler is based on the *Eclipse Modeling Framework* (EMF) (Eclipse, 2010), which provides an implementation of the *Meta Object Facility* (MOF) specification (OMG, 2006b), it is also worth remarking that the BPMN metamodel conforms to this OMG standard (see MOF at the M3 layer in Figure 1).

In our pilot project, once we had defined the workflow related to the *Incident Management* process in terms of the workflow ontology, we used this knowledge to obtain the conceptual model of the ITSMS needed to support it. For

Figure 6. The ICTD incident management process

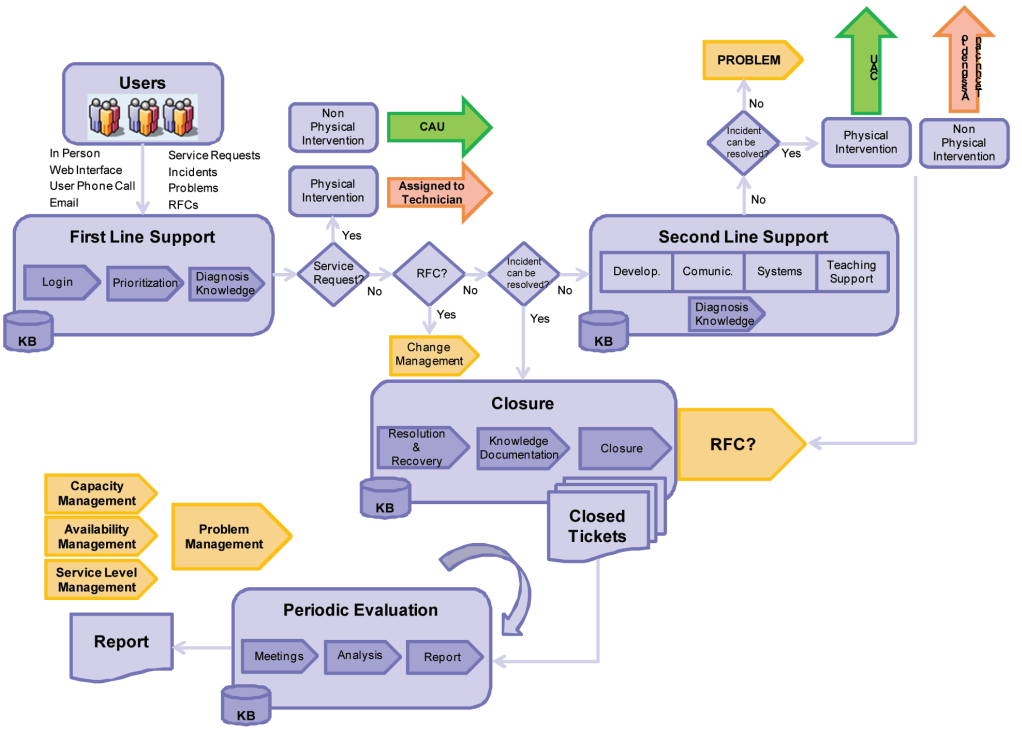


Figure 7. BPMN diagram representing the ICTD incident management process

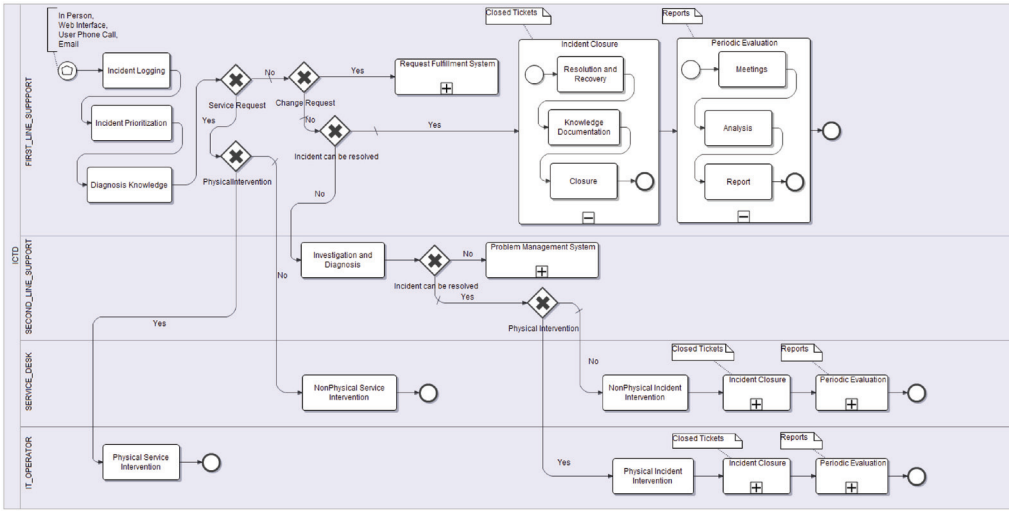
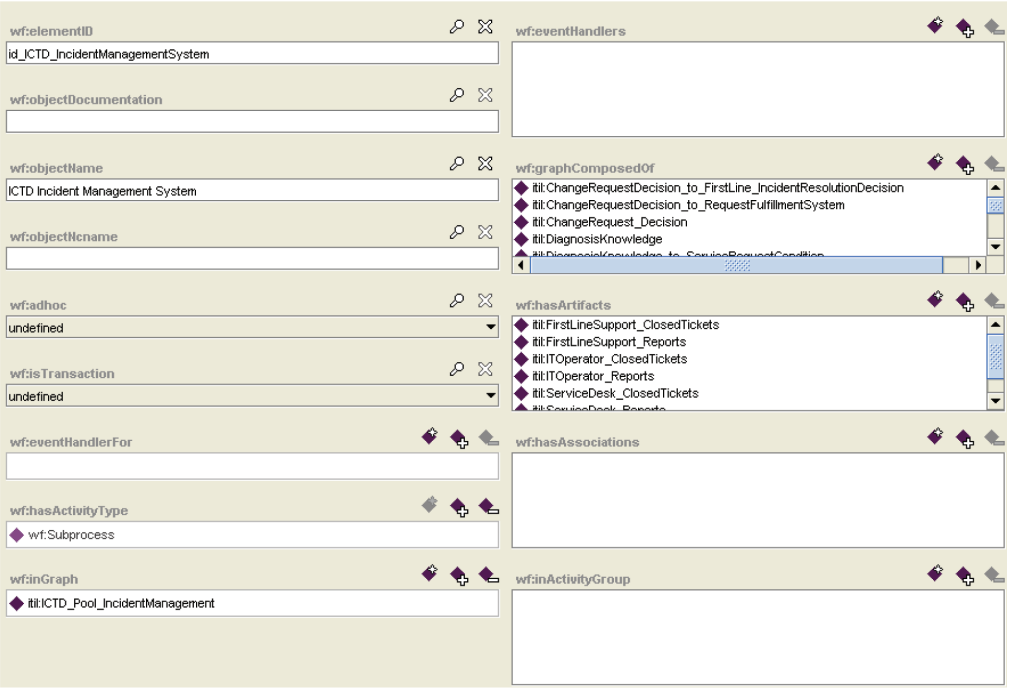


Figure 8. The itil:ICTD_IncidentManagementSystem instance



this purpose, we used a java file (*OWL2BPMN_client.java*) which: (1) presents to the user all the *itil:Activity* instances in the ontology, allowing him to select those to be automated; and (2) the *OWL2BPMNTransformer_XSLT.java* file creates a XMI-serialized ITIL model for each selected activity in terms of our ontological DSL for OWL activities (Figures 4 and 5), and generates a XMI-serialized BPMN models for the resulting OWL models with JDOM (http://www.jdom.org) by using an XSLT script (*OWL2BPMNTransformer.xslt*). Figure 9 shows an excerpt of the XSLT script that transforms a XMI file extracted from an OWL activity into the XMI file that represents a BPMN model.

Table 1 lists the mappings among OWL activity constructs and BPMN constructs. For example, in an OWL activity model (i.e., the ITIL model), the element *Activity* associated with the element *graphComposedOf* is transformed into the model element *vertices xmi:type="bpmn:Activity"* in a BPMN diagram.

4. RELATED WORK

During the recent years, there has been an increasing interest in the definition and implementation of ITSM processes, especially using ontology-based approaches. For example, Savvas and Bassiliades (2009) propose an OWL ontology that provides specific knowledge for administrative procedures.

Prieto and Lozano-Tello (2009) propose a workflow model based on ontologies to represent management processes defined in terms of workflows. The authors remark that the application of ontologies in this field can provide several advantages such as exchange of tasks and workflow model reuse. However, although the ontologies proposed by the authors can be used in the context of ITIL (their model has been used in the domain of the Incident Management process), they capture the knowledge related to workflows, not to ITSM and ITIL. As a consequence, if the users are not experts

Figure 9. Excerpt of the OWL2BPMNTransformer.xslt file

```
[...]
<xsl:template match="Workflow">
    <bpmn:BpmnDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
         xmlns:bpmn="http://stp.eclipse.org/bpmn">
        <xsl:call-template name="common-data-structure"/>
        <xsl:if test="diagramComposedOf/Pool">
            <xsl:for-each select="diagramComposedOf/Pool">
                <pools xmi:type="bpmn:Pool">
                    <xsl:call-template name="common-data-structure"/>
                    <xsl:if test="graphComposedOf/SubProcess/hasArtifacts">
                      <xsl:call-template name="artifact-structure"/>
                    </xsl:if>
                    <xsl:if test="graphComposedOf/SubProcess">
                        <xsl:if test="graphComposedOf/SubProcess/graphComposedOf">
                            <xsl:for-each select="graphComposedOf/SubProcess">
                                <xsl:call-template name="vertex-structure"/>
                                <xsl:call-template name="sequence-edge-structure"/>
                            </xsl:for-each>
                        </xsl:if>
                    </xsl:if>
                    <xsl:if test="composedOfLanes">
                        <xsl:for-each select="composedOfLanes/Lane">
                                <xsl:call-template name="lane-structure"/>
                        </xsl:for-each>
                    </xsl:if>
                </pools>
            </xsl:for-each>
        </xsl:if>
    </bpmn:BpmnDiagram>
</xsl:template>
<xsl:template name="common-data-structure">
    <xsl:attribute name="xmi:id">
        <xsl:value-of select="@xmi:id" />
    </xsl:attribute>
    <xsl:if test="elementID">
        <xsl:attribute name="iD">
                <xsl:value-of select="elementID" />
        </xsl:attribute>
    </xsl:if>
    <xsl:if test="objectName">
        <xsl:attribute name="xmi:name">
                <xsl:value-of select="objectName" />
        </xsl:attribute>
    </xsl:if>
</xsl:template>
[...]
```

Table 1. Mapping of OWL activity and BPMN constructs

OWL Activity Model	Type	BPMN Model	Type
Activity (associated with the element <graphComposedOf>)	element	vertices (with the attribute xmi:type="bpmn:Activity")	element
Activity (associated with the element <hasActivities>)	element	activities (associated with the element <lanes>)	attribute
Association	element	associations (with the attribute xmi:type="bpmn:Association")	element
DataObject	element	artifacts (with the attribute xmi:type="bpmn:DataObject")	element
elementID	element	iD	attribute
hasActivityType	element	activityType	attribute
Lane (associated with the element <composedOfLanes>	element	lanes (associated with the element <pools> and with attribute xmi:type="bpmn:Lane")	element
Lane (associated with the element <inActivityGroup>	element	lanes (associated with the element <vertices>)	attribute
objectName	element	xmi:name	attribute
Pool	element	pools (with the attribute xmi:type="bpmn:Pool")	element
SequenceEdge (associated with the element <graphComposedOf>)	element	sequenceEdges (with the attribute xmi:type="bpmn:SequenceEdge")	element
SequenceEdge (associated with the element <incomingEdges>)	element	incomingEdges (associated with the element <vertices>)	attribute
SequenceEdge (associated with the element <outgoingEdges>)	element	outgoingEdges (associated with the element <vertices>)	attribute
SubProcess	element	vertices (with the attribute xmi:type="bpmn:SubProcess")	
TextAnnotation	element	artifacts (with the attribute xmi:type="bpmn:TextAnnotation")	element
xmi:id	attribute	xmi:id	attribute
workflow	element	bpmn:BpmnDiagram	element

in ITIL, the definition of workflows using their proposal is quite complex.

For semantic annotations in business process modeling, Thomas and Fellman (2009) and Di Francescomarino et al. (2011) propose extensions of process modeling languages, such as BPMN, using a formal ontology. The semantic process modeling proposed by Thomas and Fellman uses the *Suggested Upper Merged Ontology* (SUMO) (Niles & Pease, 2001) for the ontology construction and OWL as the ontology language. In Di Francescomarino et al. (2011), the authors propose a framework for the collaborative specification of semantically annotated business processes. The proposed framework is based on the notion of a shared workspace aimed at obtaining annotated *Business Process Diagrams* (BPDs) specified using BPMN, where each BPD element is considered as an instantiation of an element specified in their BPMN Ontology (Data Knowledge and Management, 2008).

In the same context, Born, Dörr, and Weber (2007) propose an approach to the integration

of semantics in modeling tools to support the graphical modeling of business processes with information derived from domain ontologies. For this purpose, the authors propose the use of an extended BPMN ontology, called Semantic Business Process Modeling Notation (sBPMN) (Abramowicz et al., 2007). This ontology enables designers to augment and annotate business process models.

5. CONCLUSION AND FUTURE WORK

In this paper, we have defined an ontology-based and model-driven approach that helps bridging OE and MDE in the development of software systems aimed to maintain and improve IT service quality in line with business requirements. In our opinion, in the analysis phase (conceptual modeling) of any software system, the emphasis must be placed on the data (i.e., in the domain information), rather than in the operations (i.e., the behavior). In this vein, the MDA allow us to represent models of the real-world using conceptual models that abstract key domain concepts, represent them appropriately and allow us to transform them into code correctly.

Through the definition of an ontology-based ITSM model we introduce the usage of semantic information during the conceptual modeling of IT service-oriented software systems that support ITSM processes associated with an ITSMS. In this way, we formalize and describe coherently and consistently all of the knowledge related to ITSM best practices in order to ease service management, including the workflow related to its implementation. Thus, each ontology-based workflow model represents a perspective or concern of an IT service-oriented software system. The representation of the ITSM processes in terms of the workflow ontology has also the advantage of providing us with new inferred knowledge, and being machine-processable.

Using our approach, we create BPMN models at the CIM level (according to the MDA approach), from our OWL activity models, that conceptually represent the high-level requirements of the software systems needed to support one or more ITSM processes in an ITSMS. To the best of our knowledge, this is the first approach the combines OE and MDE using OWL, SWRL, and BPMN for conceptual model enrichment in the implementation of IT service-oriented software systems.

Future work should focus on lower-level model transformations and on the development of additional case studies to evaluate the approach more exhaustively.

REFERENCES

Abramowicz, W., Filipowska, A., Kaczmarek, M., & Kaczmarek, T. (2007). Semantically enhanced business process modelling notation. In *Proceedings of the Workshop on Semantic Business Process and Product Lifecycle Management*.

Bézivin, J. (2005). On the unification power of models. *Journal of Software and System Modeling*, *4*(2), 171–188. doi:10.1007/s10270-005-0079-0

Bézivin, J. (2009). *Advances in model driven engineering: Achievements and challenges*. Paper presented at the 14th Conference on Software Engineering and Databases.

Born, M., Dörr, F., & Weber, I. (2007). User-friendly semantic annotation in business process modeling. In *Proceedings of the International Workshop on Human-Friendly Service Description, Discovery and Matchmaking* (pp. 260-271).

Chandrasekaran, B., Josephson, J., & Benjamins, R. (1999). What are ontologies, and why do we need them. *Journal of IEEE Intelligent Systems*, *14*(1), 20–26. doi:10.1109/5254.747902

Clark, J. (1999). *XSL transformations (XSLT) version 1.0*. Retrieved from http://www.w3.org/TR/xslt

Clark, T., Sammut, P., & Willans, J. (2008). *Applied metamodelling: A foundation for language driven development* (2nd ed.). Sheffield, UK: Ceteva.

Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2002). Methodologies, tools and languages for building ontologies. Where is the meeting point? *Data & Knowledge Engineering*, *46*(1), 41–64. doi:10.1016/S0169-023X(02)00195-7

Data and Knowledge Management. (2008). *BPMN ontology.* Retrieved from https://dkm.fbk.eu/index.php/BPMN_Ontology

Devedžić, V. (2002). Understanding ontological engineering. *Communications of the ACM*, *45*(4), 136–144. doi:10.1145/505248.506002

Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., & Tonella, P. (2011). A framework for the collaborative specification of semantically annotated business processes. *Journal of Software Maintenance and Evolution: Research and Practice*, *23*(4), 261–295. doi:10.1002/smr.525

Eclipse. (2006). *BPMN modeler.* Retrieved from http://www.eclipse.org/bpmn/

Eclipse. (2010). *Eclipse modeling framework project (EMF).* Retrieved from http://www.eclipse.org/modeling/emf/

Eriksson, H. E., & Penker, M. (2000). *Business modeling with UML: Business patterns at work*. New York, NY: John Wiley & Sons.

Ferrario, R., & Guarino, N. (2009). Towards an ontological foundation for services science. In J. Domingue, D. Fensel, & P. Traverso (Eds.), *Proceedings of the First Future Internet Symposium* (LNCS 5468, pp. 152-169).

Gašević, D., Djurić, D., & Deved, M. (2007). On metamodeling in megamodels. In G. Engels, B. Opdyke, D. C. Schmidt, & F. Weil (Eds.), *Proceedings of the 10th International Conference on Model Driven Engineering Languages and Systems* (LNCS 4735, pp. 91-105).

Gašević, D., & Hatala, M. (2010). Model-driven engineering of service-oriented systems: A research agenda. *International Journal of Service Science, Management, Engineering, and Technology*, *1*(1), 17–32. doi:10.4018/jssmet.2010010102

Goeken, M., & Alter, S. (2009). Towards conceptual metamodeling of IT governance frameworks: Approach – Use – Benefits. In *Proceedings of the 42nd Hawaii International Conference on System Sciences* (pp. 1-10).

Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning* (pp. 601-602).

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, *43*(5-6), 907–928. doi:10.1006/ijhc.1995.1081

Hochstein, A., Zarnekow, R., & Brenner, W. (2005). ITIL as common practice reference model for IT service management: Formal assessment and implications for practice. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service* (pp.704-710).

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). *SWRL: A semantic web rule language combining OWL and RuleML.* Retrieved from http://www.w3.org/Submission/SWRL/

International Organization for Standardization. (2005a). *ISO/IEC 20000-1:2005 Information Technology – Service Management – Part 1: Specification*. Geneva, Switzerland: ISO/IEC.

International Organization for Standardization. (2005b). *ISO/IEC 20000-2:2005 Information Technology – Service Management – Part 2: Code of Practice. Geneva, Switzerland: ISO/IEC. itSMF International. (2007). Foundations of IT service management based on ITIL V3*. Zaltbommel, The Netherlands: Van Haren Publishing.

Jones, S. (2005). Toward an acceptable definition of service. *Journal of IEEE Software*, *22*(3), 87–93. doi:10.1109/MS.2005.80

Kent, S. (2002). Model driven engineering. In M. Butler, L. Petre, & K. Sere (Eds.), *Proceedings of the Third International Conference on Integrated Formal Methods* (LNCS 2335, pp. 286-298).

Knowledge Based Systems, Inc. (KBSI). (1994). *Information integration for concurrent engineering (IICE) project* (Tech. Rep. No. F33615-90-C-0012). Retrieved from http://www.idef.com/pdf/Idef5.pdf

Lichtenstein, S., Nguyen, L., & Hunter, A. (2004). Issues in IT service-oriented requirements engineering. In *Proceedings of the 9th Australian Workshop on Requirements Engineering* (pp. 176-191).

Mizoguchi, R., & Ikeda, M. (19996). *Towards ontology engineering* (Tech. Rep. No. AI-TR-96-1, I.S.I.R.).Toyonaka, Japan: Osaka University.

Nagarajan, M., Verma, K., Sheth, A., Miller, J., & Lathem, J. (2006). Semantic interoperability of web services: Challenges and experiences. In *Proceedings of the 4th IEEE International Conference on Web Services* (pp. 373-382).

Nextel, S. A. (2010). *ISO/IEC 20000 para pymes: Cómo implantar un sistema de gestión de los servicios de tecnologías de la información*. Spain: AENOR Ediciones.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems* (pp. 2-9).

OGC. (2007). *The official introduction to the ITIL service lifecycle*. London, UK: The Stationery Office.

Olivé, A. (2007). *Conceptual modeling of information systems*. Berlin, Germany: Springer-Verlag.

OMG. (2003). *MDA guide, version 1.0.1.* Retrieved from http://www.omg.org/cgi-bin/doc?omg/03-06-01

OMG. (2006a). *Business process modeling notation (BPMN), version 1.0.* Retrieved from http://bpmn.org/Documents/OMG_Final_Adopted_BPMN_1-0_Spec_06-02-01.pdf

OMG. (2006b). *Meta object facility (MOF), core specification, version 2.0.* Retrieved from http://www.omg.org/spec/MOF/2.0/PDF/

OMG. (2007). *MOF 2.0/XMI mapping, version 2.1.1.* Retrieved from http://www.omg.org/spec/XMI/2.1.1/

OMG. (2010a). *Business process model and notation (BPMN), version 2.0 (Beta 2).* Retrieved from http://www.omg.org/cgi-bin/doc?dtc/10-06-04

OMG. (2010b). *Unified modeling language (UML), superstructure, version 2.3.* Retrieved from http://www.omg.org/spec/UML/2.3/Superstructure/PDF/

Prieto, A. E., & Lozano-Tello, A. (2009). Use of ontologies as representation support of workflows oriented to administrative management. *Network and Systems Management*, *17*(3), 309–325. doi:10.1007/s10922-009-9132-6

Ruiz, F., & Hilera, J. R. (2006). Using ontologies in software engineering and technology. In Calero, C., Ruiz, F., & Piattini, M. (Eds.), *Ontologies in software engineering and software technology* (pp. 62–119). Berlin, Germany: Springer-Verlag. doi:10.1007/3-540-34518-3_2

Rule, M. L. (n. d.). *The rule markup initiative.* Retrieved from http://ruleml.org/

Savvas, I., & Bassiliades, N. (2009). A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration. *Expert Systems with Applications*, *36*, 4467–4478. doi:10.1016/j.eswa.2008.05.022

Scheer, A. W. (2000). *ARIS- business process modeling* (3rd ed.). Berlin, Germany: Springer-Verlag. doi:10.1007/978-3-642-57108-4

Schmidt, D. C. (2006). Model-driven engineering. *IEEE Computer*, *39*(2), 25–31.

Smith, M. K., Welty, C., & McGuinness, D. L. (2004). *OWL web ontology language guide.* Retrieved from http://www.w3.org/ TR/owl-guide/

Stahl, T., & Völter, M. (2006). *Model-driven software development: Technology, engineering, management*. New York, NY: John Wiley & Sons.

Talantikite, H. N., Aissani, D., & Boudjlida, N. (2009). Semantic annotations for web services discovery and composition. *Computer Standards & Interfaces*, *31*(6), 1108–1117. doi:10.1016/j.csi.2008.09.041

Thomas, O., & Fellmann, M. (2009). Semantic process modeling – design and implementation of an ontology-based representation of business processes. *Journal of Business & Information Systems Engineering*, *1*(6), 438–451. doi:10.1007/s12599-009-0078-8

Uschold, M., & Grüninger, M. (1996). Ontologies: Principles, methods, and applications. *The Knowledge Engineering Review*, *11*(2), 93–113. doi:10.1017/S0269888900007797

Valiente, M.-C., García-Barriocanal, E., & Sicilia, M. A. (in press). Applying ontology-based models for supporting integrated software development and IT service management processes. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 41*(5).

Verma, K., & Sheth, A. (2007). Semantically annotating a web service. *IEEE Internet Computing*, *11*(2), 83–85. doi:10.1109/MIC.2007.48

Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S., & Miller, J. (2005). Meteor-S WSDI: A scalable infrastructure of registries for semantic publication and discovery of web services. *Information Technology Management*, *6*(1), 17–39. doi:10.1007/s10799-004-7773-4

Vicente-Chicote, C., & Alonso, D. (2007). Tutorial: Herramientas Eclipse para desarrollo de software dirigido por modelos. *XII Jornadas de Ingeniería del Software y Bases de Datos: Actas de Talleres y Tutoriales de las Jornadas de Ingeniería del Software y Bases de Datos, 1*(8).

Workflow Management Coalition (WfMC). (1999). *Terminology and glossary* (Tech. Rep. No. WfMC-TC-1011 v3). Retrieved from http://www.wfmc.org/