Knowledge-Based Systems 28 (2012) 76–87

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Applying an ontology approach to IT service management for business-IT integration

Maria-Cruz Valiente *, Elena Garcia-Barriocanal, Miguel-Angel Sicilia

Computer Science Department, University of Alcala, Polytechnic Building, Ctra. Barcelona km. 33.6, 28871 Alcala de Henares (Madrid), Spain

ARTICLE INFO

Article history:
Received 28 July 2011
Received in revised form 1 December 2011
Accepted 2 December 2011
Available online 9 December 2011

Keywords:
Knowledge representation
Process modeling
Ontologies
ITIL
ITSM
OWL
SWRL
SQWRL

ABSTRACT

Best practice frameworks focused on the integration of business and IT, such as ITIL, help organizations create and share effective service management. However, guidelines and models are commonly specified using natural language or graphical representations, both lacking the computational semantics needed to enable their automated validation, simulation or execution. This paper presents an ontology approach, which can help service providers add semantics to their service management process models and detect semantic ambiguities, uncertainties and contradictions. The proposed ontology draws its knowledge from good practice guidance for ITSM, enabling the current business gap that exists in many IT service providers to be overcome. To do so, service management processes are formalized in terms of an ontology defined using OWL combined with SWRL and SQWRL, the latter two being used to specify constraints and infer new knowledge. Our ontology provides support for executable service models with computational semantics. SWRL rules associated with the ontology can be categorized into three groups: (1) *Model consistency*; (2) SLA *breaches*; and (3) *Proactive actions*. Such rules allow us to better manage actual service management processes which are delivered in line with business needs. Also, the resulting specifications can be shared, reused and interchanged by automated means using e-business frameworks such as ebXML.

1. Introduction

In order to enhance their competitiveness and performance in the new Internet- and technology-driven business logic, organizations must assess the efficiency and quality of their services. Business is what defines the requirements of the information systems that automate everyday activities, and therefore systems must be designed to support business processes [10]. That is, organizations should be aware of the close relationship and convergence between business and *Information Technologies* (IT). However, integration business needs and technology is still a challenging issue for the quality assurance of the services that are delivered [26]. In some cases, information systems do not meet business requirements, and as a consequence many organizations regard IT as placing a limit on their business growth [20]. Business and IT are not sharing their challenges and goals, yet they must do so as one of the resources required for *sustained competitive advantage* (SCA) [49].

As a response to this problem, *IT Service Management* (ITSM) enables the integration of business with IT in terms of services (in this paper, service is understood as 'IT service') that can be managed as another business unit. IT services are recognized as crucial, strategic, organizational assets that must be managed for business success [3].

However, the complexity of service management is still a challenging problem in industry. This is still the case even when best practice guidance for ITSM is adopted, since service management processes are specified in natural language and graphical representations, and the task of connecting these models with formal semantics and machine-processable languages is complex. That is, a humanly comprehensible description may not directly be processable by computers, while semantics associated with natural language can lead to different representations and interpretations of the terms included in a specification [44]. For example, the *Information Technology Infrastructure Library* (ITIL) defines an incident as "*an unplanned interruption to an IT service or a reduction in the quality of an IT service. Failure of a configuration item that has not yet impacted service is also an incident. For example failure of one disk from a mirror set*" [33]. But, according to this definition what exactly should be classified as 'incident' in the ITSM domain? What specific information and tasks are associated with the incident management process? Which of these tasks could be automated using a computer tool? What metrics should be included in the incident management process in order to measure it? What are the different categories for ITSM metrics? What are the *Critical Success Factors* (CSFs) in the incident management process for a

* Corresponding author.
E-mail addresses: maricruz.valiente@uah.es (M.-C. Valiente), elena.garciab@uah.es (E. Garcia-Barriocanal), msicilia@uah.es (M.-A. Sicilia).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.12.003

specific IT service provider? How are those metrics related to each CSF in the incident management process?

In this paper, we address this problem by means of integration between ontologies and ITSM best practices. We aim at translating perceptions of the 'real-world' (that is, IT service management domain) expressed in natural language and graphical representations to an ontology which is a formal representation.

The aim of the proposed ontology is to support: (i) a common and unifying framework for representing ITSM knowledge based on the ITIL V3 service lifecycle; (ii) ITSM formal taxonomy development; (iii) an ITSM metrics model; (iv) reasoning and inference capabilities; (v) *Service Level Agreement* (SLA) management; and (vi) the sharing, reuse and interchange of ITSM knowledge by using different e-business frameworks in the context of business-to-business commerce. An e-business framework is a standard that defines data structures and elements in order to standardize the exchange of electronic business data [29]. For example, the ebXML framework [31] is an example that has been standardized by means of the *Extensible Markup Language* (XML).

The proposed ontology captures best practices in service management for: (i) representing a semantic model so that organizations can understand their ITSM processes (e.g., the maturity level); and (ii) business decision-making based on an ITSM metrics model, which can be executed thanks to semantic capabilities. It is worth noting that the ITIL/ITSM terminology is described according to the *Spanish Association for Standardization and Certification*[1] (AENOR). The demonstration of our approach is illustrated with a real case study of a Spanish IT service provider that implemented the ITIL *Incident Management* process as a first step to quality improvement in the delivery of their services.

The rest of this paper is structured as follows. Section 2 covers background on ITSM and ontologies. Section 3 summarizes the process followed to define the proposed ontology on the basis of the ITIL framework. Section 4 describes the constraints, queries and inference rules that can be performed on the ontology. Related work is presented in Section 5, and finally, Section 6 concludes the paper and outlines future work.

2. Background

In this section we describe background information and fundamental issues regarding ITSM, ITIL and ontologies. The complexity involved in performing quality services is of major importance. An ontology approach opens a window for the establishment of a systematic method that enables us to implement ITIL-based processes in a straightforward and well-defined manner.

2.1. IT service management

The concept of service is understood differently depending on the domain or application area, and this leads to a certain confusion that has been explored in [23,11]. For example, *Service-Oriented Architecture* (SOA) is an approach to structuring software systems by grouping functionalities into manageable services with well-defined interfaces that can be invoked remotely, where a service represents how its consumers wish to use it [23]. Within ITSM, and throughout this paper, 'service' should be understood as an overall IT service, such as software distribution or server support [3], which is made up of a combination of people, processes and technology and should be defined in an SLA [21]. An SLA represents a formal agreement between an *IT service provider* and a *customer*. The SLA describes a level of assurance or warranty with regard to the level of service quality for each of the services delivered to business (customers). In this context, IT services can therefore be considered as commitments just as in the approach of [11]. Thus, the term 'service' does not refer to Web Services in the SOA context since these approaches are outside the scope of our work. However, it is possible to use SOA and principles to develop flexible and re-usable IT services that are common and can be shared and exploited across many different areas of business [32].

There are several well established good practice frameworks to create effective IT service management systems such as ITIL. Nowadays, ITIL is the best known and most widely accepted guidance and it has become the*de facto* standard for ITSM, providing "*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*" [34].

ITIL version 3, also known as ITIL V3, is an enhanced and consolidated framework that proposes a new approach to ITSM by considering the lifecycle of a service. Given than that ITIL V3 is the most complete and up-to-date version of this ITSM framework, and since the *Office of Government Commerce* (OGC) have announced their plans to withdraw publications and qualifications of ITIL version 2, we selected ITIL V3 for our ontology approach.

For the sake of clarity and to be concise, we have selected only one part of ITIL V3 to illustrate the work presented in this paper: the *Incident Management* process from the *Service Operation* stage [33]. We have selected this process because the *Incident Management* process is highly visible to business, and is therefore often one of the first processes to be implemented in service management projects. Also, this process is a relatively simple one with a reasonable number of associated classes and properties.

2.2. Ontologies

Ontologies [17,46] are explicit representations of a shared conceptualization, i.e., an abstract, simplified view of a shared domain of discourse. More formally, an ontology defines the vocabulary of a problem domain, and a set of constraints (axioms or rules) on how terms can be combined to model specific domains. An ontology is typically structured as a set of definitions of concepts and relations between these concepts. Ontologies are machine-processable, and they also provide the semantic context by adding semantic information to models, thereby enabling natural language processing, reasoning capabilities, domain enrichment, domain validation, etc.

Ontology Engineering (OE) is sometimes seen as the next panacea in knowledge modeling, aiming at avoiding conceptual ambiguities, advocating reuse and standardization, and serving as building blocks for more complex automated-reasoning systems [5,16].

Recently, there has been increasing interest in the use of ontologies for various aspects of ITSM [3,12,38,40]. For example, Savvas and Bassiliades [38] propose an ontology in OWL [41] that provides specific knowledge for administrative procedures, which are mapped into OWL-S service models [50]. The importance of using OE to automate and validate service process models is also highlighted in [48,43]. A formal description of the functionality of a service process is crucial for service process reuse [47], whereas a formal description of the data that the service management processes exchange is a key requirement for interoperability [28]. Furthermore, if IT service providers formally define SLAs and quality-of-service attributes, they may distinguish themselves from their competitors [4,35].

Since the inception of the Semantic Web, in which ontologies are the principal resource for integrating and dealing with online information, a new set of standards have been proposed. OWL is one such standard belonging to a family of knowledge representation languages prepared for the Semantic Web (although this language can be adopted in other domains, as we do). OWL has attained the

[1] http://www.aenor.es/.

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 XML as interchange syntax. Therefore, due to their RDF basis, OWL ontologies can be associated with any other form of information expressed on the Semantic Web which allows the integration of the resulting specifications with a variety of e-business frameworks (e.g., ebXML) and with business modeling frameworks (e.g., BPMN), in both cases using XML as the interchange syntax in order to match organizations with the same business processes.

OWL ontologies are composed of: (i) classes as sets of individuals; (ii) individuals as instances of classes (i.e., objects of the domain); and (iii) properties as binary relations between individuals. In addition, it is possible to define property domains, cardinality ranges, and reasoning rules. Some reasoning engines, such as Pellet,[2] or Jess[3] can be used to infer additional facts about the knowledge explicitly included in OWL ontologies. Reasoning in OWL can be performed at a class, property or instance level. For example, it is possible to define rules for checking class equivalence, for classifying individuals, or for computing additional property values using transitiveness. The *Semantic Web Rule Language* (SWRL) is also an OWL-related specification that extends OWL with logic-based rules [19], and the *Semantic Query-Enhanced Web Rule Language* (SQWRL) is a query language for extracting information from OWL ontologies [30]. SQWRL is based on SWRL and uses its semantic foundations as its formal underpinning which, together with SWRL rules and stored facts (knowledge base), can be executed to define constraints and infer new facts.

OWL DL is the *Description Logics* (DL) sublanguage of OWL [2]. One of the key capabilities of OWL DL is its ability to define all these classes in terms of necessary and sufficient conditions. New concepts can be defined by specifying property restrictions on existing concepts. Then, an inference engine executes the ontology and will compute the new inferred ontology class hierarchy, indicating inconsistent classes (e.g., a reasoner is able to test whether or not one class is subclass of another class).

3. Onto-ITIL: an ontology for representing a semantic model for ITSM

In this section, we discuss an approach to building Onto-ITIL, an ontology for ITSM based on the *ITIL V3 Service Management Model*. Onto-ITIL is aimed at achieving knowledge formalization of the ITSM domain. The proposed ontology provides a mechanism for managing interoperability, consistency checking and decision-making, and it also can be used as a knowledge base for ITIL-based process implementations. Our approach enables IT service providers to add semantics and constraints to the data associated with the different ITIL-based processes that underpin a business, so that they can share and reuse information in a homogeneous way.

The proposed ontology is defined by adopting OWL DL, which provides automated and efficient reasoning facilities, together with SWRL and SQWRL for reasoning, and knowledge inference and retrieval. The open source Protégé-OWL tool[4] is used in this research as an ontology editor to create the ontology. Protégé includes the *SWRLTab*, which is an extension for editing and executing SWRL rules and SQWRL queries in conjunction with Jess, a rule engine.

In order to test our approach, we started a pilot project with a Spanish IT service provider that was interested in improving the quality of the services they deliver to their customers. Their objectives were to improve customer satisfaction, efficiency and competitiveness. To do so, they recognized the need for a service management framework and they realized the importance of becoming a proactive organization fully involved in their ITSM projects.

[2] http://clarkparsia.com/pellet/.
[3] http://www.jessrules.com/.
[4] http://protege.stanford.edu/.

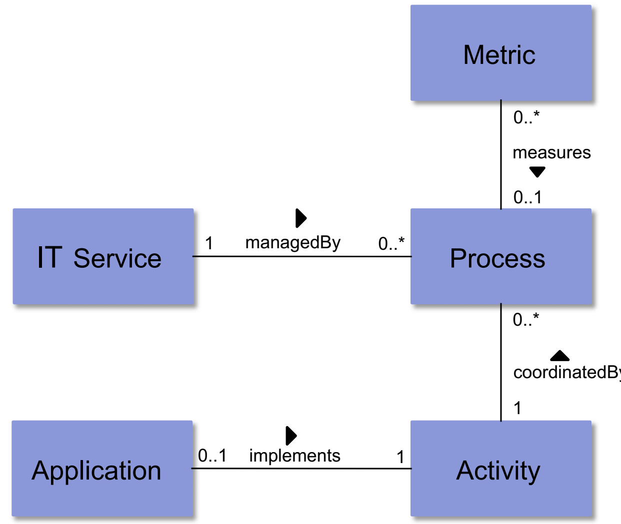

Fig. 1. Onto-ITIL principles.

3.1. Onto-ITIL principles

The semantic model for ITSM proposed in this work is based on the structure illustrated in Fig. 1, which relies on five concepts (*IT service*, *Process*, *Metric*, *Activity* and *Application*) and the four relations defined among them (*managedBy*, *measures*, *coordinatedBy* and *implements*).

In order to further detail the most relevant concepts related to the Onto-ITIL principles, some formal definitions are included in this section.

Definition 1. Let $S = \{s_1, s_2, \ldots, s_n\}$ be the *Service Portfolio*, that is, the complete set of IT services that are managed by an IT service provider. The service portfolio is a key element of ITSM and it is used to manage the entire lifecycle of each service $s_i \in S$. It includes three categories: (i) *Service Pipeline* Δ with $\Delta \subset S$ (proposed or in development); (ii) *Service Catalog* C with $C \subset S$ (live or available for deployment); and (iii) *Retired Services* R with $R \subset S$. The service portfolio represents the current contractual commitments, the development of new services, and the ongoing improvement plans initiated as part of a *Continual Service Improvement* (CSI) process.

Definition 2. An *IT service* is defined as a tuple $s_i = \{A_i, \Psi_i, M_i, Y_i\}$, where A_i represents the lifecycle of an IT service s_i, Ψ_i represents the stakeholders involved in s_i; M_i represents the set of metrics that helps manage s_i; and Y_i represents the set of applications that support s_i. As do [11], we consider IT services to be events based on agreements and modeled by a layered set of interrelated activities (events), each one with its own participants and spatiotemporal location. Therefore, IT service providers do not deliver the IT service itself, but its content, that is, "*the actions to be performed in the interest of the customer.*"

Definition 3. A *service Lifecycle* $A_i = \{\zeta_{i1}, \zeta_{i2}, \ldots, \zeta_{in}\}$ represents different stages in which an IT service s_i can be associated.

Definition 4. A *service Stage* $\zeta_{ij} = \left\{\mathrm{P}_{ij}, \Phi_{ij}^{input}, \Phi_{ij}^{output}\right\}$ represents the j stage included in the lifecycle of the IT service s_i, where P_{ij} represents the finite set of processes of the j stage required to manage s_i. Since the strength of the service management model relies on the continual feedback obtained at each service stage, Φ_{ij}^{input} represents the set of input stages that are feedback for the j stage in the lifecycle of s_i, and Φ_{ij}^{output} represents the set of output stages that receive feedback from the j stage in the lifecycle of s_i. This feedback ensures that service optimization is managed from a business perspective.

Definition 5. A *stakeholder* $\psi_i \in \Psi_i$ represents a person who has an interest in an IT Service s_i. A stakeholder may be interested in the activities, targets, resources, or deliverables. Stakeholders may include customers, partners, employees, shareholders, owners, etc.

Definition 6. A *process* $\rho_{ij} = \{\Lambda_{ij}, \mathrm{I}_{ij}, \mathrm{O}_{ij}, \mathrm{Z}_{ij}, \mathrm{K}_{ij}\}$, with $\rho_{ij} \in \mathrm{P}_{ij}$, represents an activity Λ_{ij} designed to accomplish a specific objective in the management of an IT service s_i. Each process takes one or more inputs I_{ij} and produces one or more outputs O_{ij}. Each process may have one or more interfaces Z_{ij} with other processes, and may include any number of metrics K_{ij}, with $\mathrm{K}_{ij} \subset \mathrm{M}_i$, which help measure its quality and effectiveness.

Definition 7. An *activity* $\Lambda_{ij} = \{\alpha_{ij1}, \alpha_{ij2}, \ldots, \alpha_{ijn}\}$ represents the specification of a set of actions designed to achieve a particular result for a process in the management of an IT service s_i. Actions cannot be broken down further in the activity containing them. However, the execution of a single action may induce the execution of many other actions, that is, there are actions that can also invoke activities. For example, an action that invokes an operation implemented by an activity containing actions that are executed before the invoking action is completed. In this context, an activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model.

Definition 8. A *metric* $\mu_i = \{r_1, r_2, \ldots, r_n\}$, with $\mu_i \in \mathrm{M}_i$, represents a set of measurements designed to manage an IT service s_i. A metric is a scale of measurement r_i defined in terms of a standard, for example, in terms of a well-defined unit. The quantification of an event through the process of measurement relies on the existence of explicit or implicit metrics, which are the standard to which the measurements are benchmarked.

Definition 9. An *application* $\iota_i \in Y_i$ represents a piece of software that provides the functionality required to manage an IT service s_i. Applications implement activities and each application may support one or more IT services.

Definition 10. *Application Functions* Π_i define the mapping between each activity α_{ij} and the application ι_i that supports an IT service s_i.

3.2. The Onto-ITIL ontology

In this section, we formalize the proposed semantic model using OWL. This model relies on the *ITIL V3 service management Model* and on the Onto-ITIL principles formerly described in Section 3.1. It is worth highlighting that some of the Onto-ITIL concepts have been defined in terms of other existing ontologies that gather interesting domain-independent knowledge [18]. This allows us to relate ITIL-based service management information to other data in the Semantic Web. Among the existing upper ontologies useful for defining some of the Onto-ITIL concepts, we have selected OpenCyc,[5] the public version of the Cyc technology [24]. This ontology represents the most general and complete knowledge base and reasoning engine. OpenCyc provides us with the mechanisms to define the core elements of our semantic model and to make assertions about those elements. Model elements of ITIL-based specifications are provided by separate parts of the ontology. This enables a clear separation of the different ITSM concerns and improves understanding and reusability of Onto-ITIL concepts. From here on, we use the prefixes ‘oc’ and ‘itil’ to refer to the namespaces of OpenCyc and Onto-ITIL respectively. Fig. 2 shows a general overview of the semantic model for ITSM as defined by the Onto-ITIL ontology. For reasons of space, the complete Onto-ITIL ontology is neither depicted nor described in details. The rest of this section will focus on the most important Onto-ITIL concepts represented in Fig. 2.

The architecture of our Onto-ITIL ontology is based on a service lifecycle (see Definition 3 in Section 3.1). The stages of a service lifecycle are comprised of processes (modeled using the *itil:hasProcess* property). As previously stated (see Definition 4 in Section 3.1), the strength of an ITSM model relies on the continual feedback obtained at each service stage [34]. We use the *itil:isFeedback* and *itil:receivesFeedback* properties to express the inputs and outputs provided and required at each stage.

3.2.1. Processes

In the Onto-ITIL ontology, an *itil:Process* is defined as a subclass of an *oc:ProgramSpecification*,which is in turn defined as an abstract characterization of how a program should behave. An *itil:Process* represents an activity designed to accomplish a specific objective (see Definition 6 in Section 3.1). In this context, processes are composed of activities that describe the specification in terms of workflows enriched with ontological knowledge (modeled using the *itil:specifiesActivity* property). For example, in our pilot project, the *itil:Printing_Servers_IM_Process* element was created as an instance of the *itil:IncidentManagement* concept (subclassing from *itil:OperationProcess*) that specifies the *itil:ICTD_IM_Activity* activity (instance of *itil:Activity*). In order to describe the semantics associated with these workflows, we have defined a separate ontology which is imported by the Onto-ITIL ontology. Given that *Business Process Model and Notation* (BPMN) is considered the *de facto* standard for business process specification [36], we have defined our workflow ontology (referenced by the prefix ‘wf’) in terms of the BPMN metamodel included in the BPMN Modeler project [8]. This will allow the seamless integration of our workflow specifications into the Eclipse platform [9]. The workflow ontology is not presented in detail for reasons of space.

Processes are measured in terms of metrics. In our approach, we include the complete metrics model suggested by Steinberg [42] which can be used with our semantic model for ITSM. In general, an *itil:Metric* (see Definition 8 in Section 3.1) is a scale of *itil:Measurement* defined in terms of a standard, i.e., in terms of a well-defined unit (modeled using the *itil:includesMeasurement* property). Each metric has a type (modeled using the *itil:MetricType* enumeration class) and they must be designed in line with customer (business) requirements for ITSM.

3.2.2. Events

An *oc:Action* is the super class for all the concrete action types defined in Onto-ITIL. All actions are performed by an *oc:Agent-Generic*, i.e., the stakeholder responsible (modeled using the *oc:performedBy* property). The *oc:PurposefulAction* concept is the

[5] http://www.opencyc.org/.

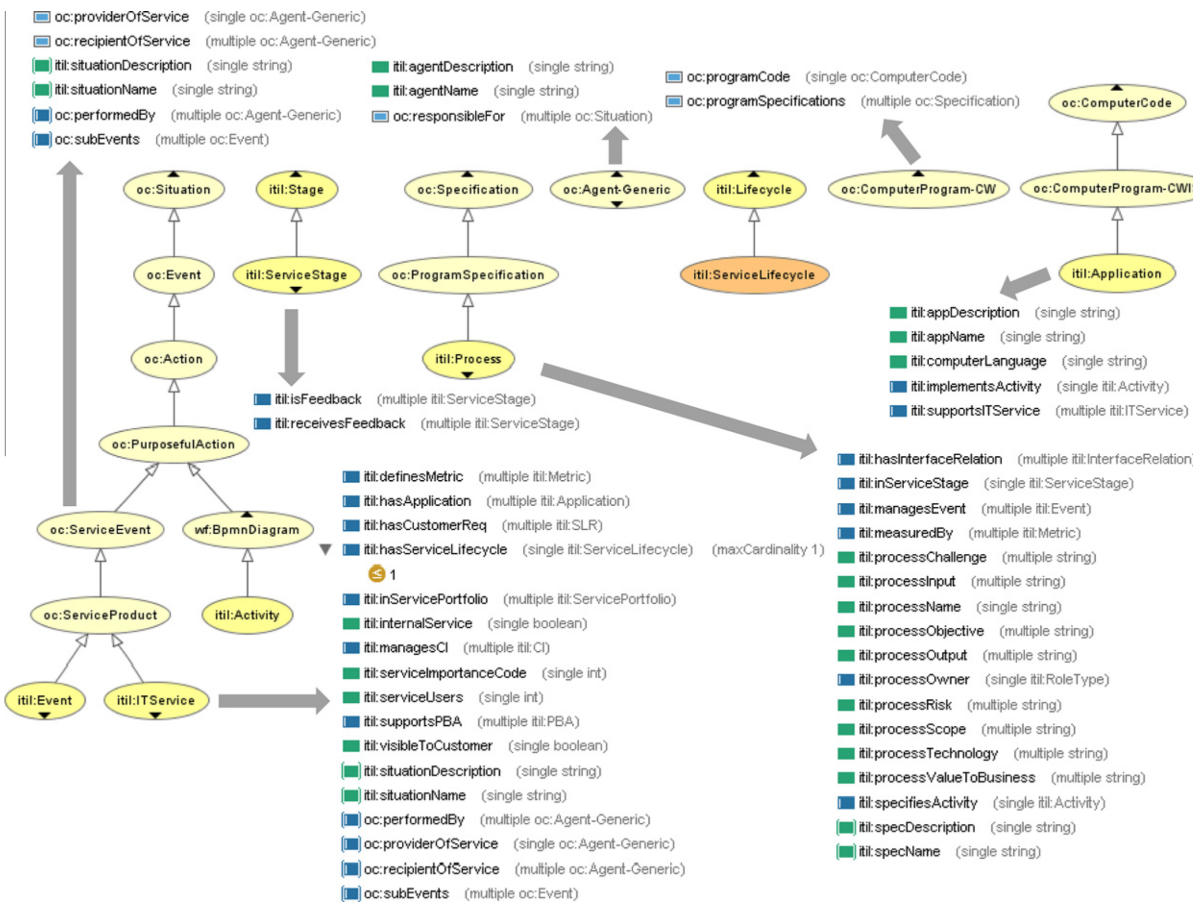

Fig. 2. The structure of the Onto-ITIL ontology.

subclass of *oc:Action*. An *oc:PurposefulAction* is used in our approach to classify the activities involved in an ITIL-based workflow process (i.e., the set of events, the order in which they must be performed, and the stakeholders that participate in the process), and to classify service events associated with our semantic model. In Onto-ITIL, *wf:BpmnDiagram* and *oc:ServiceEvent* concepts are subclasses of *oc:PurposefulAction*.

The *wf:BpmnDiagram* is the workflow representation (i.e., the workflow model) in the form of a BPMN diagram composed of pools and messages. In our approach, we consider *itil:Activity* (see Definition 7 in Section 3.1) a subclass of *wf:BpmnDiagram* with a view to modeling the high level requirements of the information system that could automate the activities defined as part of a workflow associated with an ITSM model. For example, in our pilot project, an instance of *itil:Activity*, *itil:ICTD_IM_Activity*, specifies the workflow that defines the tasks to be carried out when an incident is reported and is related to a specific incident management process instance, such as *itil:Printing_Servers_IM_Process* (modeled using the *itil:specifiesActivity* property).

An *oc:ServiceEvent* represents the super class for all concrete events. An *oc:ServiceProduct* is an *itil:ServiceEvent* done for payment. In our approach, *itil:Event* and *itil:ITService* are subclasses of *oc:ServiceProduct*.

An *itil:Event* is any detectable or discernible occurrence that has significance for the management of the IT infrastructure or the delivery of an IT service, and for the evaluation of the impact a deviation might cause to the services. We use the *itil:Event* class to specify all events that are included in an IT service for proactive and reactive event management. According to ITIL, some events could be part of different processes, or even a combination of two or more of them. Therefore an IT service provider must decide and indicate what process (or processes) is going to manage a specific event (modeled using the *itil:managesEvent* property). In our proposal, *itil:Incident*, *itil:ServiceRequest*, *itil:RFC*, *itil:Change* and *itil:Problem* are the subclasses of *itil:Event*. Also, activities undertaken to manage a specific event are included using the *itil:undertakesActivity* property. For example, in our pilot project, *itil:Teaching_Server_Failure* (instance of *itil:Incident*) defines the characteristics of this kind of managed event in the organization, the actions to be performed in order to resolve it (modeled using the *itil:undertakesActivity* property) and, since it is an incident associated with the incident management process of the *Printing Server* service, the incident is related to the *itil:Printing_Servers_IM_Process* instance (modeled using the *itil:managesEvent* property).

3.2.3. IT services

An *itil:ITService* (see Definition 2 in Section 3.1) is an *oc:ServiceProduct* provided to one or more customers by an IT service provider (see Definition 5 in Section 3.1). That is, IT services represent the means of delivering value to customers by facilitating outcomes which, since they are based on agreements, have to be defined in an SLA. The *itil:CoreService* and *itil:SupportingService* concepts are the subclasses of *itil:ITService*. An *itil:CoreService* represents a service that delivers the basic outcomes desired by the customer. Core services represent the value that the customer wants and for which they are willing to pay. Core services anchor the value proposition for the customer and provide the basis for their continued utilization and satisfaction. For example, in our pilot project, *itil:Printing_Servers*, *itil:DNS_Service*, *itil:Staff_email*, *itil:HW_Management* and *itil:Software_Licensing* are examples of instances of *itil:CoreService*. An *itil:SupportingService* is a service that enables or enhances a core service. For example, *itil:Backup* and *itil:Mailing_Lists* instances. These two classes (*itil:CoreService* and *itil:SupportingService*) are related to each other through the *itil:hasSupportingService* property.

The *itil:ServicePortfolio* (see Definition 1 in Section 3.1) is the complete set of IT services (modeled using the *itil:detailsITService* property) that are managed by an IT service provider. The *itil:ServicePortfolio* is used to manage the entire lifecycle of all services and includes three categories (modeled using the *itil:ServicePortfolioType* enumeration class): *itil:SERVICE_PIPELINE*, *itil:SERVICE_CATALOG* and *itil:RETIRED_SERVICES*. For example, in our pilot project *itil:ICTD_ServiceCatalog* is an instance of *itil:ServicePortfolio*, where the *itil:hasServicePortfolioType* property is equal to *itil_SERVICE_CATALOG*, and is related to different instances of *itil:ITservice* through the *itil:detailsITService* property.

3.2.4. Applications

An *itil:Application* is a piece of software that provides the functionality required by an *itil:ITService* (see Definition 9 in Section 3.1). According to Definition 10 in Section 3.1, each *itil:Application* implements an *itil:Activity* (modeled using the *itil:implementsActivity* property) and may be part of one or more *itil:ITService* (modeled using the *itil:supportsITService* property). For example, in our pilot project the *itil:HEAT_Help_Desk_Software* element was created as an instance of the *itil:Application* concept that currently implements the *itil:ICTD_IM_Activity* activity (coordinated by incident management processes) and supports the *itil:Printing_Servers* service (among others).

3.2.5. Service level agreements

For SLA management (Fig. 3) we have included the OpenCyc concept *oc:Contract*, defined as a legal agreement in which two or more agreeing stakeholders promise to do (or not do) something. The OpenCyc concept *oc:LegalAgreement* represents a specialization of *oc:Agreement*. An *oc:Agreement* involves two or more parties who agree on certain propositions (modeled using the *oc:agreeingAgents* property). Making the propositions true may require some action or commitment of resources on the part of one or more of the agreeing stakeholders. In our context, an *oc:Contract* is composed of one or more *oc:ContractDocument* (modeled using the *itil:agreesContractDocument* property).

An *itil:SLA* represents the *itil:Agreement* (subclass of *oc:ContractDocument*) that describes a formal understanding of an agreement between customers and IT service providers. That is, an SLA is a written agreement between an IT service provider and the customer, defining the key service targets and responsibilities of both parties. Each agreement defines a business process that enables the delivery of a service (modeled using the *itil:definesBusinessProcess* property). An SLA describes the service and service level targets, and specifies the responsibilities of the IT service provider and the customer (modeled using *itil:CustomerRelation* and *itil:ITServiceProviderRelation* classes). An SLA represents the level of assurance or warranty with regard to the level of service quality delivered by an IT service provider to its customers for each of the IT services delivered to business. In our pilot project, the *itil:SLAIncidentResolution* class is used to specify the agreed incident resolution times for customers in a specific SLA. Also, SLAs are related to *Operational Level Agreements* (OLAs) and *Underpinning Contracts* (UCs), which provide support for SLA fulfillment (modeled using *itil:OLA* and *itil:UC* classes, and *itil:supportedByOLA* and *itil:supportedByUC* properties). The *itil:OLA* is an agreement between an IT service provider and a third party that assists with the provision of services to customers. However, in this case the third party is another part of the same organization. The *itil:OLA* defines the goods or services to be provided and the responsibilities of

Fig. 3. The structure of the Onto-ITIL SLA knowledge.

Table 1
Mapping between ebXML constructs and Onto-ITIL constructs.

ebXML construct	Onto-ITIL construct
ebxml:MultipartyCollaboration	wf:Pool, wf:Lane and itil:RoleType
ebxml:BusinessPartnerRole	itil:RoleRelation
ebxml:Performs	oc:performedBy
ebxml:AuthorizedRole	oc:IntelligentAgent, oc:responsibleFor and itil:RoleRelation
ebxml:BinaryCollaboration	wf:Activity
ebxml:BusinessTransactionActivity	wf:Activity and wf:ActivityType=”SUBPROCESS” OR “TASK”
ebxml:CollaborationActivity	wf:Activity and wf:ActivityType=”SUBPROCESS” OR “TASK”
ebxml:BusinessTransaction	wf:Activity and wf:ActivityType=”SUBPROCESS” OR “TASK”
ebxml:RequestingBusinessActivity	wf:Activity and wf:ActivityType=”TASK”
ebxml:RespondingBusinessActivity	wf:Activity and wf:ActivityType=”TASK”
ebxml:DocumentEnvelope	oc:Agreement
ebxml:BusinessDocument	oc:Agreement
ebxml:Transition	wf:Association and wf:SequenceEdge
ebxml:Start	wf:Activity and wf:ActivityType=”EventStartEmpty” OR ”EventStartMessage” OR “EventStartMultiple” OR “EventStartRule” OR ”EventStartTimer” OR “EventStartLink” OR “EventStartSignal”
ebxml:Sucess	wf:Activity and wf:ActivityType=”EventEndEmpty” OR ”EventEndMessage” OR “EventEndCompensation” OR “EventEndTerminate” OR ”EventEndLink” OR “EventEndMultiple”
ebxml:Failure	wf:Activity and wf:ActivityType=”EventEndError”
ebxml:Fork	wf:Activity and wf:ActivityType=”GatewayDataBasedExclusive” OR “GatewayEventBasedExclusive” OR “GatewayDataBasedInclusive” OR ”GatewayParallel” OR “GatewayComplex”
ebxml:Join	wf:Activity and wf:ActivityType=”GatewayDataBasedExclusive” OR “GatewayEventBasedExclusive” OR “GatewayDataBasedInclusive” OR ”GatewayParallel” OR “GatewayComplex”

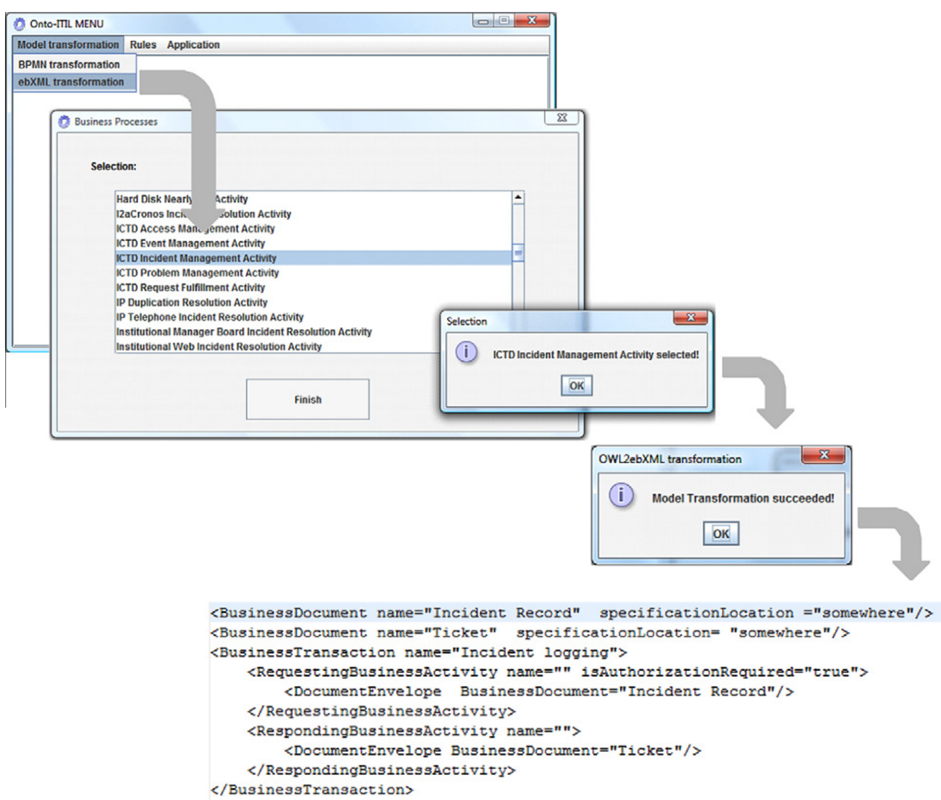

Fig. 4. Sample of ebXML model transformation (excerpt from the *ICTD_IM_Activity.ebxml* file).

both parties. For example, there could be an OLA between the IT service provider and a procurement department to obtain hardware in agreed times. Finally, the *itil:UC* is an agreement between an IT service provider and a third party. In this case, the third party (supplier) is another organization. The UC defines targets and responsibilities that are required in order to meet agreed service level targets in an SLA.

3.2.6. Mapping ebXML with ITIL

Since suppliers (internal or external) and the management of suppliers and partners are essential to the provision of quality IT services [32], we can obtain the internal and cross-organizational integration of the supporting services through the management of OLAs and UCs using ebXML business process specifications. The Onto-ITIL concepts *itil:OLA* and *itil:UC* represent the *Collaboration Protocol Agreements* (CPAs) established between business parties in the ebXML domain. This means that both parties carry out electronic business directly according to a specific CPA (i.e., the IT service provider and its supplier follow the business process defined in the CPA). For example, in our pilot project a new computer tool for incident management was required in order to implement *itil:Printing_Servers_IM_Process*. Therefore, the *itil:ICTD_IM_Activity* business process, an instance of *itil:Activity* that specifies the corresponding process flow, needed to be transformed into an ebXML model and associated with the *itil:ICTD_IM_OLA* instance which represented the CPA document.

The resulting mapping between ebXML business process specifications constructs [45] and Onto-ITIL constructs for supplier management is summarized in Table 1 (ebXML abstract classes and optional classes have been omitted). As can be seen in this table, some ebXML constructs are derived from the combination of other constructs in the Onto-ITIL model. In order to validate our approach, we implemented a prototype in Java with the Eclipse platform that generates the transformation from an Onto-ITIL model to an ebXML model: (i) all business processes instantiated in the ontology are presented to users, allowing them to select those to be transformed; and (ii) the OWLTransformer_ebXML.java file creates an ebXML model for each selected business process. For example, in our pilot project the output transformation of the *Incident Logging* action (instance of *wf:Activity*), which represents a business activity (transaction) that is part of the *itil:ICTD_IM_Activity* business process, is shown in Fig. 4.

4. Rule-based constraints and knowledge inference

In our approach, we use SWRL for consistency checking, model validation, business rules analysis, decision-making, etc. The set of rules defined in SWRL are combined with an OWL ontology. This relation provides all the relevant aspects of an ITIL-based specification and improves the management of IT services. In order to test our approach with regard to reasoning capabilities and rule chaining, in our prototype these rules are executed in Java using the Jess rule engine (Fig. 5). This enables us to verify constraints and inconsistencies in our semantic model, and incorporate new knowledge into the ontology, thereby improving IT service management. We consider three types of rules (*model consistency*, *SLA breaches* and *proactive actions*) which are detailed in the following subsections.

4.1. Model consistency

Model consistency rules are applied to all the instances included in Onto-ITIL models. Here are some examples of model consistency rules. In our pilot project the following rule states that it is possible to have the same activity related to different ITIL-based processes, but of the same type:

```
itil:Activity(itil:ICTD_IM_Activity)
∧itil:Process(?p)∧
itil:coordinatedByProcess(itil:ICTD_IM_Activity,?p)
→
itil:IncidentManagement(?p).
```

This rule states that if an incident management activity (*a*) is coordinated by a specific process (*p*), then *p* must represent an incident management process (i.e., instances of the *itil:IncidentManagement* class). Inconsistencies are detected and a list of processes that do not meet this SWRL rule definition is displayed to users (Fig. 6).

Similarly, the next rule defined states that if a KPI is related to a specific process then, given that a KPI is a metric that enables business decisions in the delivery of a service, it must be a metric belonging to the IT service associated with the process:

```
itil:ITService(?serv)∧itil:ServiceLifecycle(?l)∧
itil:hasServiceLifecycle(?serv,?l)∧
itil:ServiceOperation(?st)∧
itil:inServiceLifecycle(?st,?l)∧
itil:OperationProcess(?p)∧
itil:hasOperationProcess(?st,?p)∧itil:KPI(?m)
∧itil:measures(?m,?p)
→
itil:definesMetric(?serv,?m).
```

In this case, if an IT service (*serv*) has a service lifecycle (*l*), a service operation stage (*st*) is part of *l*, an operation process is one of the processes included in *st*, and *m* is a KPI that measures *p*, then *serv* must have *m* as a defined metric too.

The next rule shows how it is possible to force the computation of a specific metric in order to document it and test its results following the metrics model proposed in [42]:

```
itil:OperationalMetric
(itil:Total_number_of_incidents)∧itil:OperationalMetric
(itil:Number_of_incidents_resolved_within_
    agreed_service_levels)
itil:metricValue(itil:Total_number_of_incidents,?v1)∧
itil:metricValue(itil:Number_of_incidents_
    resolved_within_agreed_service_levels,?v2)∧
itil:KPI(itil:Incident_resolution_rate)
∧swrlb:divide(?ratio,?v2,?v1)∧
swrlb:multiply(?result,?ratio,100)
→
itil:metricValue(itil:Incident_resolution_rate,?result),
```

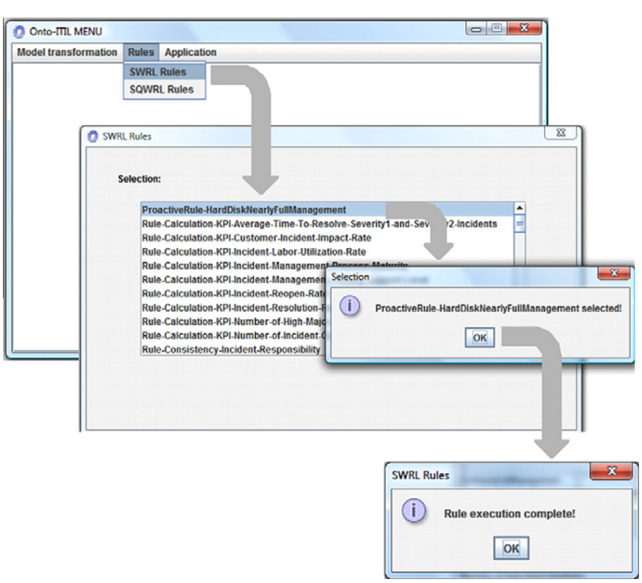

Fig. 5. Rule selection and execution.

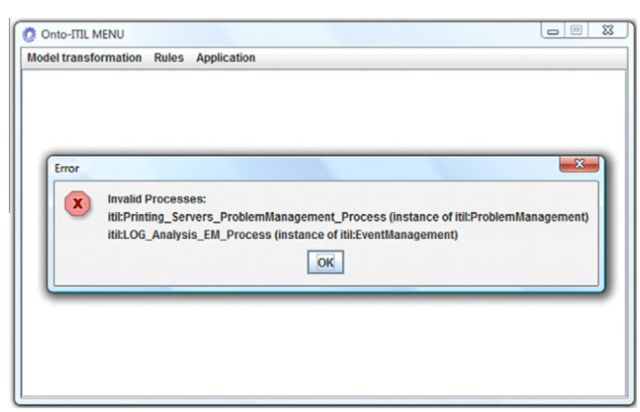

Fig. 6. SWRL rule validation.

where, the KPI *k* associated with the incident resolution rate is defined as the ratio (*result*) between the number of incidents resolved within the agreed service levels and the total number of incidents.

As a final example, the following SQWRL query extracts the list of incidents associated with each customer group managed by a specific IT service provider as part of its *Incident Management* process for the *Printing Servers* service. The results of this query can help IT service providers decide whether or not the incidents have been properly assigned and managed:

```
itil:Incident(?i)∧itil:IncidentManagement
(itil:Printing_Servers_IM_Process)∧
itil:managesEvent(itil:Printing_Servers_IM_Process,?i)∧
itil:situationName(?i,?name)∧
itil:hasIncidentRecord(?i,?r)∧
itil:incidentPriority(?r,?pr)∧
itil:hasIncidentGroup(?r,?gr)∧
itil:incidentPriority(?r,?pr)
→
sqwrl:select(?name,?gr,Numberofincidents)∧
sqwrl:count(?r)∧
sqwrl:columnNames(Name,Priority,Description,Count).
```

4.2. SLA breaches

SLA breaches are rules that check whether the agreed level of assurance or warranty regarding the level of service quality achieved by IT service providers for each of the services delivered to their customers is met. For example, in our pilot project the priority of an *incident* is used to obtain the maximum resolution time agreed. Therefore, we define the following SWRL rule for assigning an agreed resolution time (hours) to a specific customer

```
itil:CoreService(itil:Printing_Servers)∧
itil:ServiceLifecycle(itil:Printing_Servers_
      ServiceLifecycle)∧
itil:hasServiceLifecycle(itil:Printing_Servers,
itil:ICTD_ServiceLifecycle)∧
itil:SLA(itil:SLA_CUSTOMER_1)∧
itil:coveringITService(itil:SLA_CUSTOMER_1,
itil:Access3G)∧
itil:SLAIncidentResolution(itil:CUSTOMER_1_
      SLAIncidentResolution_10)
→
itil:hasSLAIncidentResolution(itil:SLA_CUSTOMER_1,
itil:CUSTOMER_1_SLAIncidentResolution_10)∧
itil:slaIncidentPriority(itil:CUSTOMER_1_
SLAIncidentResolution_10,10)∧
itil:slaIncidentResolutionTime(itil:CUSTOMER_1_
      SLAIncidentResolution_10,12).
```

In this case, the SLA for a specific customer (*itil:SLA_CUSTOMER_1*) in the service *itil:Printing_Servers* states that for an incident of priority 10, the maximum resolution time is 12 h. In addition, the priority of a specific incident is calculated according to the following rules:

```
itil:ITService(?serv)∧
itil:serviceImportanceCode(?serv,?code)∧
itil:ServiceLifecycle(?l)∧
itil:hasServiceLifecycle(?serv,?l)∧
itil:ServiceOperation(?st)∧
itil:inServiceLifecycle(?st,?l)∧
itil:IncidentManagement(?p)∧                      (1)
itil:hasOperationProcess(?st,?p)∧
itil:Incident(?i)∧itil:managesEvent(?p,?i)∧
itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)
→
itil:incidentUrgency(?r,?code)
```

```
itil:ITService(?serv)∧itil:serviceUsers(?serv,?usr)∧
itil:ServiceLifecycle(?l)∧
itil:hasServiceLifecycle(?serv,?l)∧
itil:ServiceOperation(?st)∧
itil:inServiceLifecycle(?st,?l)∧
itil:IncidentManagement(?p)∧
itil:hasOperationProcess(?st,?p)∧
itil:Incident(?i)∧itil:managesEvent(?p,?i)∧
itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)
→
itil:incidentImpact(?r,?usr)
                                                  (2)
```

```
itil:Incident(?i)∧itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)∧
itil:incidentUrgency(?r,?u)∧
itil:IncidentGroupType(itil:GOVERNANCE)∧          (3)
itil:hasIncidentGroup(?r,itil:GOVERNANCE)
→
itil:incidentLevel(?r,?u)
```

```
itil:Incident(?i)∧itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)∧
itil:incidentUrgency(?r,?u)∧
itil:IncidentGroupType(itil:STAFF)∧
itil:hasIncidentGroup(?r,itil:STAFF)∧             (4)
swrlb:equal(?u,3)
→
itil:incidentLevel(?r,2)
```

```
itil:Incident(?i)∧itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)∧
itil:incidentLevel(?r,?l)∧
itil:incidentImpact(?r,?imp)∧
swrlb:equal(?l,5)∧swrlb:greaterThan(?imp,10000)
→
itil:incidentPriority(?r,10)
                                                  (5)
```

```
itil:Incident(?i)∧itil:IncidentRecord(?r)∧
itil:hasIncidentRecord(?i,?r)∧
itil:incidentLevel(?r,?l)∧
itil:incidentImpact(?r,?imp)∧
swrlb:equal(?l,0)∧swrlb:greaterThan(?imp,10000)
→
itil:incidentPriority(?r,5)
                                                  (6)
```

This is an example of rule chaining, where rule (**1**) calculates the incident urgency from the level of importance (*code*) of the affected IT service (*serv*). Then, rule (**2**) calculates the incident impact from the number of users (*usr*) of the affected service (*serv*). Rules (**3**) and (**4**) calculate the level of an incident (i) from the incident urgency (u) and from the type of group (g) that reported the incident. For example, if the incident has been reported by the 'GOVERNANCE' group, then the incident level must be equal to the incident urgency, but if the incidence has been reported by the 'STAFF' group, then the incident level could be less than the incident urgency. Finally, rules (**5**) and (**6**) make use of the incident impact (*imp*) and the incident level (l), respectively, to assign the incident priority. The organization states that the impact, urgency and priority codes range from 0 to 10, with 10 representing the highest priority. After rule execution, users can check incident records associated with IT services as shown in Fig. 7.

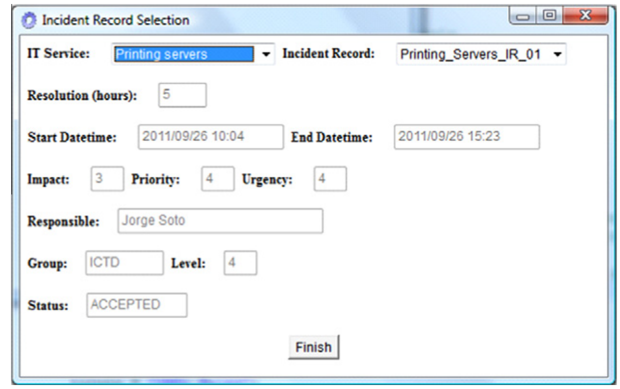

Fig. 7. Sample of an incident record after rule execution.

4.3. Proactive actions

Proactive actions are rules aimed to help organizations define how to act in order to prevent possible service failures that may occur in the future. The following example from Jerphanion and Kristelijn [22], describes a situation requiring a proactive action: "*An IT employee observes that the central hard disks are nearly full. He knows that this will lead to service failure in the near future, which will generate incidents. To prevent these incidents from occurring and to make sure that the service will remain available, the IT employee takes actions.*" According to ITIL, proactive actions are defined as part of one (or a combination of) different processes. Proactive action when a nearly full hard disk is detected can be expressed in SWRL rule as follows:

$$
\begin{aligned}
&\mathtt{itil:Event(itil:Hard\_Disk\_Nearly\_Full)}\wedge\\
&\mathtt{itil:IncidentManagement}\\
&\mathtt{(itil:Department\_Servers\_IM\_Process)}\wedge\\
&\mathtt{itil:IncidentManagement}\\
&\mathtt{(itil:File\_Servers\_IM\_Process)}\wedge\\
&\mathtt{itil:IncidentManagement}\\
&\mathtt{(itil:Implementation\_and\_Deployment\_IM\_Process)}\wedge\\
&\mathtt{itil:IncidentManagement(itil:Backup\_IM\_Process)}\\
&\rightarrow\\
&\mathtt{itil:hasEventType(itil:Hard\_Disk\_Nearly\_Full,}\\
&\mathtt{itil:WARNING)}\wedge\\
&\mathtt{itil:hasManagedEventType(itil:Hard\_Disk\_Nearly\_}\\
&\mathtt{Full,itil:PROACTIVE\_PASSIVE)}\wedge\\
&\mathtt{itil:managesEvent(itil:Department\_Servers\_IM\_}\\
&\mathtt{Process,itil:Hard\_Disk\_Nearly\_Full)}\wedge\\
&\mathtt{itil:managesEvent(itil:Files\_Servers\_IM\_Process,}\\
&\mathtt{itil:Hard\_Disk\_Nearly\_Full)}\wedge\\
&\mathtt{itil:managesEvent(itil:Implementation\_and\_}\\
&\mathtt{Deployment\_IM\_Process,}\\
&\mathtt{itil:Hard\_Disk\_Nearly\_Full)}\wedge\\
&\mathtt{itil:managesEvent(itil:Backup\_IM\_Process,}\\
&\mathtt{itil:Hard\_Disk\_Nearly\_Full)}
\end{aligned}
$$

In our pilot project, the event of a nearly full hard disk is managed by instances of incident management processes from different IT services: *itil:Department_Servers*, itil:*Files_Servers_IM_Process*, *itil:Implementation_and_Deployment* and *itil:Backup*. Also, the event is considered as a warning event, the monitoring and control of which is proactive and passive.

5. Related work

To date, there have been several proposals that apply ontologies to modeling and managing the knowledge of specific domains taking into account the heterogeneous and ambiguous nature of some of them [13,27,25,1,7]. For example, Gaeta et al. [13] define OWL ontologies in order to obtain tailored e-Learning experiences, and Moss et al. [27] use a knowledge base consisting of several OWL ontologies as key components in monitoring and treating patients who are often critically ill and may deteriorate rapidly. All these works show the benefits and potential uses of ontologies to knowledge management improvement.

In recent years there has been a lot of research regarding business models based on ontologies for representing knowledge of business processes [37,44,14]. For example, Prieto and Lozano-Tello state that the application of ontologies in this field brings with it several advantages such as an exchange of tasks and workflow model reuse. However, although these ontologies can be used in the context of ITIL (their model has been used in the domain of the Incident Management process), they only capture the knowledge related to workflows, not to ITSM or ITIL in general, as it the intention of our approach. Consequently, as the authors explain, if the users are not experts in ITIL, the definition and design of workflows using their proposal is complex. On the other hand, Garcia-Crespo et al. propose an ontology-based process model representation in order to enable the automation of e-business processes in general. Although their approach can be used for representing other business areas, it is not so much concerned with IT service quality improvement and ITIL-based process development as our approach.

In the context of ITSM, Freitas et al. [12] propose a generic ontology for IT Services in terms of UML models, and using OCL for the rules. One drawback to this approach when compared with our solution is that UML [36] is an object-oriented graphical notation for general purpose system modeling and not a process-oriented modeling language. Since the UML semantics is expressed using natural language, there may be some problems of misinterpretation and imprecision in knowledge reuse and sharing. Another drawback is that UML is a general-purpose language and therefore, even when the aim is to create a generic specification, a UML profile is needed in order to capture the ITSM knowledge in a well-defined manner. Using an OWL ontology, our approach captures the *ITIL V3 Service Management Model*, enabling users to focus on and reuse specific ITIL process implementations and to obtain automated reasoning.

Lastly, Ghedini and Gostinski [15] propose a framework using ontologies to provide business-IT alignment. In order to build the ontologies, they use ITIL V2 to obtain concepts related to ITSM using a subset of vocabulary from a business domain ontology related to the largest public bank in Brazil. The proposed framework assists in the concrete realization of governance models in the sense of understanding the effects generated between business and IT purposes, but their work is not focused on implementations of the ITIL processes. As a consequence, the ITIL ontology is simplified and, unlike the ontology we propose, it does not meet all (key) aspects related to ITSM that are defined in ITIL (e.g., metrics and documents).

6. Conclusions

In this paper, we have proposed Onto-ITIL, an ITIL-based ontological approach focused on the implementation of processes in the context of ITSM, which is aimed at easing the integration of business information and IT. Our work captures the knowledge of best practice guidance in ITSM, including the workflow specifications and the rest of ITIL-based IT service management model elements, such as metrics and formal documents. In many IT service providers there is a lack of tools and automation for reporting on metrics. Therefore, Onto-ITIL includes an ITSM metrics model that enables IT professionals to establish an ITSM metrics program in order to demonstrate the ITIL value or to operate in a cycle of continuous service improvement. The Onto-ITIL metrics model includes the metrics that should actually be measured and used for each ITIL process, thereby providing a basis for making decisions relevant to business.

The Onto-ITIL principles separate the ITIL specification part from the organization specification part, as well as from the software implementation part, of an IT service management model in order to manage their evolution and coordinate the actual ITIL-based activities associated with all the organization's specifications. Different instances of Onto-ITIL classes have been created according to the *ITIL V3 Service Management Model*. These instances can be copied, changed and adapted to the specific needs of IT service providers, while still remaining in accordance with ITIL V3.

OWL DL is used to represent our proposed ontology, and SWRL and SQWRL are used to obtain the set of rules that allows us to specify the underlying constraints and to infer new knowledge. The Protégé tool and a Java application are used to implement our proposal. We have grouped the set of SWRL rules into three different categories that will be used as the basis for defining service management rules. The combination of ontology-based and rule-based reasoning capabilities is used: (1) to validate the ITIL-based service management model (*model consistency*), (2) to manage SLA specifications (*SLA breaches*), and (3) to detect and prevent future problems/incidents (*proactive actions*). To the best of our knowledge, such a formal ITIL-based service management model definition has not been proposed for the existing ontologies in ITSM.

To validate our proposed approach, we set up a pilot project with a Spanish IT service provider. We implemented the ITIL Incident Management process as a starting point for the improvement of their service delivery. The *itil:Printing_Servers_IM_Process*, instance of *itil:IncidentManagement* class was defined in accordance with ITIL, allowing them to benefit from: (1) detection and resolution of incidents in a specific IT service which results in lower downtime to business, which in turn means higher availability of the service; (2) increased productivity through the quick solution of customer queries and incidents; (3) guidance on root causes, such as poor user training, through effective reporting; (4) alignment of IT activities to real-time business priorities; (5) identification of potential improvements to the service (the result of the establishment of a metrics model and of understanding what constitutes an incident, as well as of being in contact with the activities of business operational staff); and (6) during handling of incidents, identification of additional service or training requirements found in IT or business.

The benefits of our approach are twofold: (i) we can define the semantics and constraints associated with IT service management avoiding ambiguities and enabling a better and common understanding of ITSM processes; and (ii) we can overcome the gap that exists in many organizations between business needs and IT services delivered to customers. As a consequence, the resulting semantic-enriched specifications can be shared, interchanged and reused in a variety of XML-based e-business frameworks and modeling frameworks, such as ebXML.

These organization specifications allow more reuse and innovation in a company without affecting compliance with the best practices in ITSM. Also, the tailored workflows of the ITIL-based processes, defined as instances in our ontology, can be used as a semantic assistant of the high-level requirements elicitation for the supporting information systems in ITSM. For example, the resulting models could represent the domain modeling (the OE part) in a model-driven software development process [39]. Just as Devedzic remarked, if ontologies are not used, different knowledge representations of the same domain could be incompatible even if they use similar knowledge models for the implementation of the related software systems [6].

In summary, in this paper we show ITIL-based processes formalized by means of our proposed ontology, which can help IT service providers adopt and adapt best practices in ITSM in a consistent and a well-defined manner for business success.

References

[1] Y. Afacan, H. Demirkan, An ontology-based universal design knowledge support system, Knowledge-Based Systems 24 (4) (2011) 530–541.
[2] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider, Description Logic Handbook, Cambridge University Press, 2003.
[3] J. Black, C. Draper, T. Lococo, F. Matar, C. Ward, An integration model for organizing IT service management, IBM Systems Journal 46 (3) (2007) 405–422.
[4] J. Cardoso, A. Sheth, J. Miller, J. Arnold, K. Kochut, Quality of service for workflows and Web service processes, Journal of Web Semantics 1 (3) (2004) 281–308.
[5] B. Chandrasekaran, J. Josephson, R. Benjamins, What are ontologies, and why do we need them, IEEE Intelligent Systems 14 (1) (1999) 20–26.
[6] V. Devedzic, Understanding ontological engineering, Communications of the ACM 45 (4) (2002) 136–144.
[7] H. Dong, F.K. Hussain, Semantic service matchmaking for digital health ecosystems, Knowledge-Based Systems 24 (6) (2011) 761–774.
[8] Eclipse, 2011a. BPMN Modeler website. <http://www.eclipse.org/bpmn/>
[9] Eclipse, 2011b. <http://www.eclipse.org>
[10] H.E. Eriksson, M. Penker, Business Modeling with UML: Business Patterns at Work, Wiley, New York, 2000.
[11] R. Ferrario, N. Guarino, Towards an ontological foundation for services science, in: J. Domingue, D. Fensel, P. Traverso (Eds.), FIS 2008, LNCS, vol. 5468, Springer, Heidelberg, 2009, pp. 152–169.
[12] J. Freitas, A. Correia, F. Brito e Abreu, An ontology for IT services, in: Proceedings of the 13th Conference on Software Engineering and Databases, 2008. Available at: <http://ctp.di.fct.unl.pt/QUASAR/Resources/Papers/2008/freitas2008JISBD.pdf>.
[13] M. Gaeta, F. Orciuoli, P. Ritrovato, Advanced ontology management system for personalised e-Learning, Knowledge-Based Systems 22 (4) (2009) 292–301.
[14] A. Garcia-Crespo, B. Ruiz-Mezcua, J.L. Lopez-Cuadrado, I. Gonzalez-Carrasco, Semantic model for knowledge representation in e-business, Knowledge-Based Systems 24 (2) (2011) 282–296.
[15] C. Ghedini, R. Gostinski, A methodological framework for business-IT alignment, in: Proceedings of the Third IEEE/IFIP International Workshop on Business-Driven IT Management (BDIM), 2008, pp. 1–10.
[16] T. Gruber, The role of common ontology in achieving sharable, reusable knowledge bases, in: J.A. Allen, R. Fikes, E. Sandewall (Eds.), Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, Cambridge, 1991, pp. 601–602.
[17] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human–Computer Studies 43 (5-6) (1995) 907–928.
[18] N. Guarino, Formal ontology and information systems, in: Proceedings of FOIS'98, Trento, Italy, IOS Press, Amsterdam, 1998.
[19] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission. Available at: <http://www.w3.org/Submission/SWRL/>.
[20] ISO, 2005. ISO/IEC 20000:2005 Information Technology – Service Management.
[21] itSMF, ITIL V3: Glossary of Terms and Definitions, 2007. Available at: <http://www.itsmfi.org/files/ITILV3_Glossary_English_v1_2007_0.pdf>.

[22] S. Jerphanion, I. Kristelin, Applying the Five Disciplines of the Learning Organization to ITIL, Van Haren Publishing, 2008.
[23] S. Jones, Toward an acceptable definition of service, IEEE Software 22 (3) (2005) 87–93.
[24] D.B. Lenat, Cyc: a large-scale investment in knowledge infrastructure, Communications of the ACM 38 (11) (1995) 33–38.
[25] C-L. Liu, CDADE: conflict detector in activity diagram evolution based on speech act and ontology, Knowledge-Based Systems 23 (6) (2011) 536–546.
[26] X.F. Liu, L. Zhu, Design of SOA based web service systems using QFD for Satisfaction of Quality of Service Requirements, in: Proceedings of the 2009 IEEE International Conference on Web Services, 2009, pp. 567–574.
[27] L. Moss, D. Sleeman, M. Sim, M. Booth, M. Daniel, L. Donaldson, C. Gilhooly, M.m. Hughes, J. Kinsella, Ontology-driven hypothesis generation to explain anomalous patient responses to treatment, Knowledge-Based Systems 23 (4) (2010) 309–315.
[28] M. Nagarajan, K. Verma, A. Sheth, J. Miller, J. Lathem, Semantic Interoperability of Web services: challenges and Experiences, in: Proceedings of the 4th IEEE Int'l Conf. Web Services, IEEE CS Press, 2006, pp. 373–382.
[29] J.M. Nurmilaakso, EDI, XML and e-business frameworks: a survey, Computers in industry 59 (4) (2008) 370–379.
[30] M.J. O'Connor, A. Das, SQWRL: a query language for OWL, in: OWL: Experiences and Directions (OWLED 2009), Fifth International Workshop, Chantilly, VA, 2009.
[31] OASIS, 2001. Business process and business information analysis overview. version 1.0 (ebXML). Available at: <http://www.ebxml.org/specs/bpOVER.pdf>
[32] OGC, ITIL Service Design, TSO (The Stationery Office), London, 2007a.
[33] OGC, ITIL Service Operation, TSO (The Stationery Office), London, 2007b.
[34] OGC, The Official Introduction to the ITIL Service Lifecycle, TSO (The Stationery Office), London, 2007c.
[35] N. Oldham, K. Verma, A. Sheth, F. Hakimpour, Semantic WS-agreement partner selection, in: Proceedings of the15th Int'l World Wide Web Conf. (WWW2006), ACM Press, 2006, pp. 697–706.
[36] OMG, Business Process Model and Notation (BPMN) Version 2.0, 2010a. Available at: <http://www.omg.org/cgi-bin/doc?dtc/10-06-04>.
[37] A.E. Prieto, A. Lozano-Tello, Use of ontologies as representation support of workflows oriented to administrative management, Journal of Network and Systems Management 17 (3) (2009) 309–325.
[38] I. Savvas, N. Bassiliades, A process-oriented ontology-based knowledge management system for facilitating operational procedures in public administration, Expert Systems with Applications 36 (2009) 4467–4478.
[39] D.C. Schmidt, Model-driven engineering, IEEE Computer 39 (2) (2006) 25–31.
[40] Z. Shangguan, Z. Gao, K. Zhu, Ontology-based process modeling using eTOM and ITIL, in: IFIP International Federation for Information Processing, vol. 255, Research and Practical Issues of Enterprise Information Systems II, vol. 2, 2007, pp. 1001–1010.
[41] M.K. Smith, C. Welty, D.L. McGuinness, OWL Web Ontology Language Guide, W3C Recommendation, 2004. Available at: <http://www.w3.org/TR/owl-guide/>.
[42] R.A. Steinberg, Measuring ITIL: Measuring, Reporting and Modeling – The IT Service Management Metrics That Matter Most to IT Senior Executives, Trafford Publishing, 2006.
[43] H.N. Talantikite, D. Aissani, N. Boudjlida, Semantic annotations for Web services discovery and composition, Computer Standards & Interfaces 31 (6) (2009) 1108–1117.
[44] O. Thomas, M. Fellmann, Semantic process modeling – design and implementation of an ontology-based representation of business processes, Business and Information Systems Engineering (BISE) 1 (6) (2009) 438–451.
[45] UN/CEFACT and OASIS, 2001. ebXML Business Process Specification Schema. v1.01.
[46] M. Uschold, M. Grüninger, Ontologies: principles, methods, and applications, Knowledge Engineering Review 11 (2) (1996) 93–113.
[47] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, J. Miller, Meteor-S WSDI: a scalable infrastructure of registries for semantic publication and discovery of Web services, Journal of Information Technology and Management 6 (1) (2005) 17–39.
[48] K. Verma, A. Sheth, Semantically annotating a Web service, IEEE Internet Computing 11 (2) (2007) 83–85.
[49] M. Wade, J. Hulland, Review: the resource-based view and information systems research: review, extension, and suggestions for future research, MIS Quarterly 28 (1) (2004) 107–142.
[50] World Wide Web Consortium (W3C). OWL-S: Semantic Markup for Web Services. W3C Member Submission, 2004. Available at: <http://www.w3.org/Submission/OWL-S/>.