Using an Ontology-based Approach for Integrating Applications to support Software Processes

Conference Paper · October 2018

DOI: 10.1145/3275245.3275269

Monalessa Perini Barcellos
Universidade Federal do Espírito Santo

Ricardo de Almeida Falbo
Universidade Federal do Espírito Santo

Using an Ontology-based Approach for Integrating Applications to support Software Processes

Laylla D.C. Renault
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espírito Santo /UFES
Vitoria - Brazil
laylladuarte@gmail.com

Monalessa Perini Barcellos
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espirito Santo /UFES
Vitoria - Brazil
monalessa@inf.ufes.br

Ricardo de Almeida Falbo
Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department
Federal University of Espirito Santo /UFES
Vitoria - Brazil
falbo@inf.ufes.br

ABSTRACT

Software organizations use several applications to support their software processes. To properly support the software processes, applications should be integrated at different layers (data, service, and process). Moreover, the integration should cover semantic aspects. Therefore, an approach that provides guidelines on how to perform integration at different layers addressing semantic aspects can be helpful. This paper presents an extension of the Ontology-based Approach for Semantic Integration (OBA-SI), focusing on semantic integration at process layer. This extension establishes relationships between integration at data, service and process layers, and uses task ontologies and a process ontology to guide integration at process layer. It was used to provide an integrated solution involving applications supporting the Issue Management and Software Configuration Management processes.

CCS CONCEPTS

• Software and its engineering → Systems Integration

KEYWORDS

Ontology, system integration, process integration, semantics

ACM Reference format:
Laylla D. C. Renault, Monalessa P. Barcellos and Ricardo A. Falbo. 2018. Using an Ontology-based Approach for Integrating Applications to Support Software Processes. In *Proceedings of Simpósio Brasileiro de Qualidade de Software (SBQS'18).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3275245.3275269

SBQS, October 17–19, 2018, Curitiba, Brazil

ACM ISBN 978-1-4503-6565-9/18/10…$15.00
https://doi.org/10.1145/3275245.3275269

[1] In this paper, *application* and *system* are used as synonymous.

1 Introduction

Software organizations often use different applications[1] to support different processes. For example, schedule and budget applications are used to support project management; modeling applications are used to support requirements engineering, and development environments and version control systems are used to support coding and source code management.

Software processes are related to each other [1]. For instance, the Requirements Engineering process is related to the Testing process, because outcomes of the first are input for the second (e.g., requirements and use cases can be used to develop test cases). Therefore, software processes should be performed in an integrated way. Moreover, the different applications used to support the software processes should be integrated to properly support the software processes.

Semantic conflicts occur when applications use different meanings to the same information item, i.e., when information items seem to have the same meaning, but they do not. Neglecting semantic conflicts in Application Integration (AI) can lead to integrated solutions that fail in achieving their purposes. To reduce these conflicts, AI should address semantic issues. In this context, ontologies can be used as an interlingua to map the concepts used by different applications, enabling data, service and process understanding [3].

AI can occur in three layers [2]: data, message/service, and process. To better support the processes involved in an AI initiative, integration at process layer should be addressed. It aims at creating a choreography engine that orchestrates data and message exchange between applications, resulting in a workflow to better support the processes. It is very important for software organizations because often applications are built considering part of the processes and should be integrated to support an entire process or a set of related processes [4].

Semantic AI is not a trivial task. Thus, a systematic approach guiding such endeavors is helpful. Calhau and Falbo [3] proposed the Ontology-Based Approach for Semantic Integration (OBA-SI). OBA-SI deals with integration at data, service and process layers by using domain ontologies to assign semantics to applications' structural and behavioral conceptual models. A premise of OBA-SI is that semantics assignment must be independent of the integrated solution itself. Thus, OBA-SI focuses on assigning semantic to the elements to be shared and establishing a semantic agreement between the applications at the conceptual level (i.e., before implementation).

Although OBA-SI [3] defines steps to guide semantic AI and addresses integration at data, service and process layers, it defines only generic activities that can be applied to any of the three layers. These activities do not deal with particularities about what should be done in each layer. Moreover, the semantic integration is based only on domain ontologies. Since domain ontologies describe domain concepts and relations [5] they are enough to data integration and are helpful to assign semantics to service parameters. But, they are not enough to properly support integration at service and process layers (mainly at the last one), because services and processes have behavioral aspects that are not properly addressed by domain ontologies.

Considering that (i) software processes should be performed in an integrated way; (ii) AI is needed to better support software processes; (iii) semantic aspects must be addressed in AI initiatives; (iv) AI must address the process layer to properly support the software processes; (v) OBA-SI provides guidelines to support semantic AI, but is limited when addressing the process layer, we decided to extend OBA-SI by detailing the semantic integration activities to each layer and making explicit the connections between the integration layers. To do so, we use task ontologies to guide integration at service and process layers, since task ontologies describe the conceptualization related to a generic task or process [5] and are more suitable to assign semantics to services and processes than domain ontologies. Moreover, we developed a Business Process Ontology to help integration at process layer. After evolving OBA-SI, we used it to integrate applications to support in an integrated way the Issue Management and Software Configuration Management processes.

In this paper, we present the OBA-SI extension and its use to integrate software process supporting applications. Section 2 provides the background for the paper. Section 3 describes the followed research method. Section 4 introduces the Business Process Ontology. Section 5 concerns OBA-SI extension. Section 6 regards OBA-SI use. Section 7 discusses related works and Section 7 presents final considerations.

2 Background

The lack of integration between the applications adopted to support software processes requires software engineers to manually create and manage the relations between the processes, causing inactive management, redundancies, and inconsistencies [6]. AI is crucial to assure the traceability of artifacts and to support the automation of critical software processes [7]. However, each application tends to stand alone and depend on its own private way of structuring data, services and processes [8].

AI can be performed at three layers [2]: data, message, and process. *Data integration* deals with moving or federating data between multiple data stores. It assumes bypassing the application logic and manipulating data directly in the database. *Message (or service) integration* addresses messages exchange between the integrated applications. *Process integration*, in turn, views enterprises as a set of interrelated processes and it is responsible for handling message flows, implementing rules and defining the overall process execution. It constitutes the most complex integration approach and, different from data and service integration, process integration is often not explicitly defined [4].

Semantic conflicts can occur in any layer, arising whenever applications are built based on different conceptualizations. To avoid them, the meaning of the interchanged information has to be shared between the applications. Ontologies can be used to establish a common understanding about the universe of discourse, serving as an interlingua for communication between the applications [9]. Ontologies can be classified in [5]: *foundational ontologies*, which describe very general concepts that are independent of a particular task or domain; *domain* and *task ontologies*, which describe, respectively, the vocabulary related to a generic domain or a generic task or process; and *application ontologies*, which describe concepts depending both on a particular domain and task.

Taking into account the benefits of using ontologies to avoid semantic problems in AI, Calhau and Falbo developed OBA-SI (Ontology-Based Approach for Semantic Integration) [3]. OBA-SI approaches the integration process similarly to the software development process, containing phases related to analysis, design, implementation, testing, and deployment. OBA-SI focuses on the first phase. Semantics is addressed during the first phase. Domain ontologies are used to establish semantics at data and service layers. At data layer, concepts and relations of the applications' conceptual models (previously retrieved) are mapped to the domain ontologies' concepts and relations. These mappings are said vertical mappings (VMs) and aim to assign meaning to the applications' elements by relating them to elements in the ontology. Based on the VMs, an integration model is built, so that each element of the integration model has a meaning. In the service layer, inputs and outputs of the applications' behavioral models (also previously retrieved) are mapped to the ontology concepts. These mappings are used to support identifying related services. However, semantics is not truly assigned to messages/services neither to processes.

3 Research Method

The research method adopted in this work followed the Design Science Research paradigm. According to [13], a design science research comprises three related cycles.

In the Relevance Cycle, the problem to be addressed is defined. The problem addressed by this work regards the need for solving semantic conflicts in AI efforts, considering, besides the data layer, the service and process integration layers. The problem was identified from the literature and it was also perceived from practical experiences with OBA-SI. From a systematic mapping [10], we noticed that semantic AI initiatives at the process layer have been carried out without following a systematic approach. Moreover, they do not consider a general conceptualization about processes neither use task ontologies. From practical use of OBA-SI, we noticed that: domain ontologies are not enough to properly assign semantics to services and processes [11]; task ontologies can help in this matter [12] [13]; and it is not clear how to perform integration activities at service and process layers, due to the lack of details on what needs to be done to perform integration at each layer. Considering OBA-SI limitations identified in practice and the gaps we found in the literature, we decided to extend OBA-SI aiming to improve integration at service and process layers. To do that, we use task ontologies to address semantics at service and process layers and provide specific guidelines for performing integration analysis in those layers. Moreover, we foresee that integration at process layer should consider a common understanding about process. Although the cited practical experiences with OBA-SI have occurred in the software process context, OBA-SI is not limited to integrate applications in that domain. Contrariwise, OBA-SI can be used to integrate applications in any domain. Thus, the common understanding about process should not be limited to software processes. It should cover business processes, which involves software processes and also other organizational processes. Therefore, the common understanding about process can be provided by a business process ontology.

The Design Cycle concerns the development and evaluation of artifacts or theories to solve the identified problem. In this work, the developed artifact is a new version of OBA-SI. This new version was applied to integrate applications supporting the Issue Management and Software Configuration Management processes.

Finally, the Rigor Cycle refers to knowledge use and generation. In this work, the main foundations are knowledge obtained from the systematic mapping [10] and knowledge related to semantic AI and ontologies. The main contributions are the new version of OBA-SI and the integrated solution developed using OBA-SI. Secondary contributions are the systematic mapping [10] and the Business Process Ontology.

4 Business Process Ontology

Aiming at providing a shared and explicit description of the conceptualization related to business process, we developed the Business Process Ontology (BPO). The main intended use of BPO is to support AI at process layer. Thus, currently, it focuses on business process definition, not addressing business process execution. BPO was developed following the Systematic Approach for Building Ontologies [14]. Aligned to the characteristics of beautiful ontologies [15], BPO is based on a foundational ontology (the Unified Foundational Ontology – UFO [16]) and on business process literature (mainly [17]). Moreover, it is modular, being divided into three sub-ontologies: Business Process Goals and Types, which addresses organization's goals, business processes types and relations between them; Business Processes and Activities, dealing with the definition of business processes and their activities; and Business Process Supporting Enterprise Applications, treating applications, applications services and their support to business processes and activities. Figure 1 shows a fragment of BPO. Discussions regarding grounding BPO in UFO are outside the scope of this paper. Thus, UFO concepts are not shown in the figure.

A **Business Process** is a plan (a complex action universal) that can be described in a **Business Process Definition Document** recognized by at least one **Organization**. For instance, the *Software Project Management* business process is described in the *Software Project Management Process Specification* document, which is recognized by the *Organization O*. A Business Process can be decomposed into other Business Processes (said its **sub-processes**) or into ***Business Process Activities***, which, in turn, can be decomposed into **sub-activities,** and can depend on other Business Activities, creating the notion of **pre-activity** and **post-activity**. For example, *Project Planning* and *Project Monitoring and Control* could be sub-processes of the *Software Project Management* business process and *Develop the Project Plan* could be a business activity of *Project Planning*.

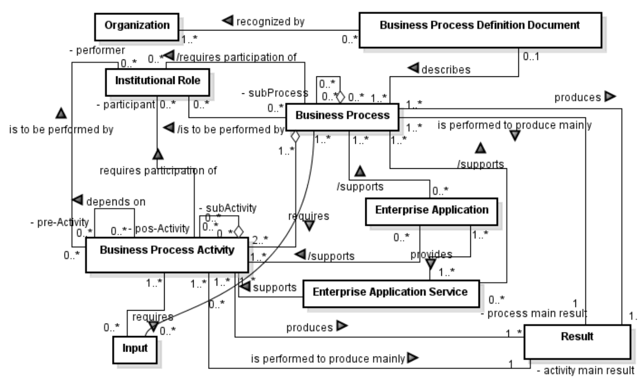

Figure 1: BPO fragment

Business activities and business processes can require **Inputs** (e.g., *Client Information* is an input for the *Develop the Project Plan* business activity) and produce **Results** (e.g., *Project Plan* is a result of the *Develop the Project Plan* business activity). Among the results produced by a business process or activity, there is a result characterized as the main one (**process main result** and **activity main result**). For example, the *Project Planning* business process produces several results, such as the *Project Requirements Document* and the *Meeting Report*, but its main result is the *Project Plan*.

Business activities and processes are to be performed by **Institutional Roles** said the process or activity **performers** (e.g., the *Develop the Project Plan* activity could be performed by a

Project Manager). Business activities and processes can also require institutional roles acting as **participants** (e.g., *Client* is a participant in the *Project Planning* process). Finally, business activities and processes can be supported by **Enterprise Applications** (e.g., *DotProject*) and **Enterprise Application Services** provided by them (e.g., the *Activity's Dates Definition* service provided by *DotProject*).

5 Extending OBA-SI

In this section, we discuss the main improvements made in OBA-SI. Figure 2 presents an overview of OBA-SI current version's integration process. The symbol indicates that the referred phase or activitiy is further decomposed into other activities. The improvements made in the integration process concentrate on the *Integration Analysis* phase.

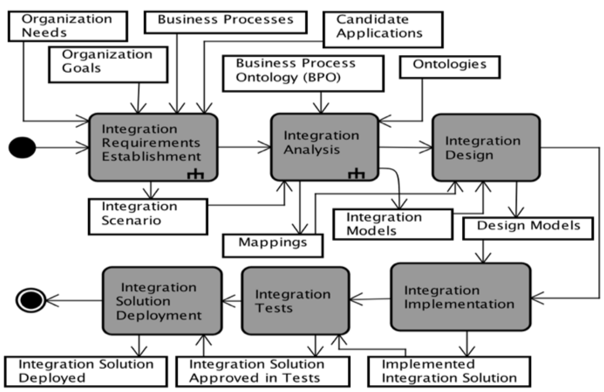

Figure 2: OBA-SI integration process

The OBA-SI integration process starts with the establishment of the integration requirements, by following a goal-oriented approach. Based on the organization goals and needs, the integration goals are defined. Thus, the business processes and domains involved in the integration, and the business activities to be supported by the integration solution are identified. Moreover, the applications to be integrated are selected and the non-functional requirements to be met by the solution are established. The information produced during the integration requirements establishment is recorded in the Integration Scenario, which is an input for the Integration Analysis phase.

As discussed before, in OBA-SI [3], the Integration Analysis phase was decomposed into generic activities that could be applied to any of the three integration layers. Consequently, particularities related to performing those activities at each layer were not explored. Moreover, the connections between the integration layers were not established. Aiming to address this issue, in the current version, we split the generic activities into specific ones dealing with the particularities of each layer. By doing that, the activities to be performed at each layer and their relations are clear. Additionally, taking the integration scenario to be addressed into account, the integration analyst can perform only the activities related to the integration layers that s/he wants to deal with, as long as dependencies between the layers are respected. For instance, to completely address integration at service layer, it is necessary to perform data integration, because services use data as inputs/outputs to provide their functionalities. Analogously, to completely address integration at process layer, service integration must be carried out, because services are used to achieve process integration. Figure 3 shows an overview of the Integration Analysis phase. Activities in gray are to be performed regardless the layers to be addressed in the solution; the one in blue refers to integration at data layer; the ones in green concern integration at service layer, and those in yellow address integration at process layer.

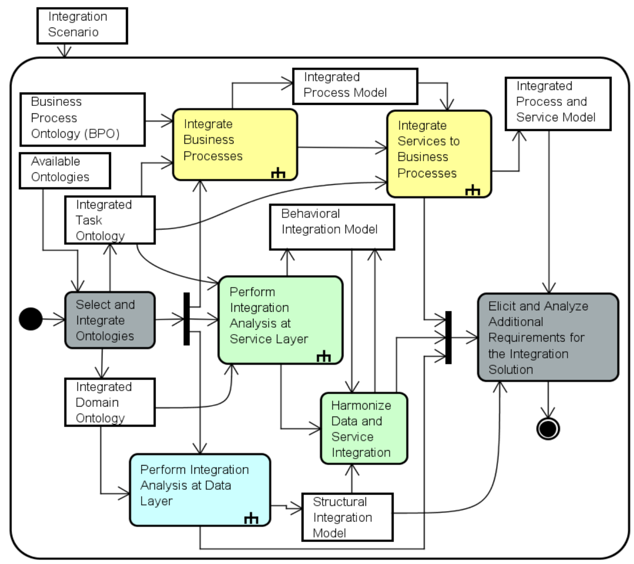

Figure 3: Integration Analysis phase

As Figure 3 shows, OBA-SI uses domain and task ontologies in different activities. Domain ontologies are used at both data and service layers. At data layer, domain ontologies' concepts, relations and properties are used to assign semantics to classes, associations and attributes of the applications' data conceptual models. For addressing integration at service and process layers, OBA-SI uses task ontologies. At service layer, domain ontology concepts are used to assign semantics to inputs and outputs of the functionalities/services provided by the applications being integrated. Task ontologies, in turn, are used to assign semantics to the applications' functionalities/services and to the business processes' activities. Thus, analogous to the structural mappings between the model elements of the applications' structural models and the concepts and relations of the domain ontology, semantics can be assigned by mapping functionalities/services and process activities to activities of the task ontology.

Since ontologies are an essential ingredient to OBA-SI, Integration Analysis starts with the selection of the ontologies to be used to assign semantics to the applications' elements. The ontologies must describe the domains and processes established in the Integration Scenario. If only data integration is to be addressed, only domain ontologies are required. Otherwise, to address service and process layers, task ontologies are also required. When selecting the ontologies, it is also necessary to identify their fragments relevant to the integration. If the necessary ontologies are not available, they must be developed. It is possible that more than one domain or task ontology are necessary to cover the domains or processes involved in the integration initiative. In this case, the ontologies must be integrated in a way that results in a single domain ontology and a single task ontology to be used in the integration initiative. Moreover, when both domain and task ontologies are needed, they must be integrated, giving rise to an application ontology, involving both domain and task perspectives. The outputs of the Select and Integrate Ontologies activity are an Integrated Domain Ontology and an Integrated Task Ontology.

After obtaining the ontologies, activities related to integration analysis at each layer can be performed. As shown in Figure 3, from data to process integration, the result of integration analysis at a certain layer is used as an input to the integration analysis at the above layer. Once integration analysis for each layer is done, it is possible to identify process activities that are not supported by any of the integrated applications. If it is desired to extend the support to the integrated process, new functionalities/services should be incorporated into the integrated solution. The *Elicit and Analyze Additional Requirements for the Integrated Solution* activity aims at eliciting and analyzing requirements for new functionalities/ services to be addressed by the integration solution.

5.1 Integration at Data Layer

The *Perform Integration Analysis at Data Layer* activity has as main result the Structural Integration Model, which is the overall conceptual model of the data involved in the integration initiative. This activity is the same as in the first version of OBA-SI [3]. As Figure 4 shows, this activity is decomposed into four activities: *Retrieve Structural Conceptual Models*, *Perform Structural Vertical Mappings*, *Develop Structural Integration Model*, and *Perform Structural Horizontal Mappings*. The Integrated Domain Ontology is used as the interlingua to perform the structural semantic mappings (vertical mappings). The Structural Integration Model includes the already harmonized elements (classes, attributes, and relations) related to data provided by the applications being integrated. Horizontal mappings relate the applications' model elements to the elements of the Integration Model.

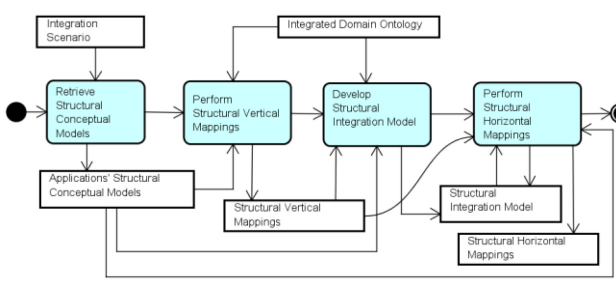

Figure 4: Integration Analysis at Data Layer

5.2 Integration at Service Layer

Integration at service layer is addressed in two steps. Firstly, in the *Perform Integration Analysis at Service Layer* activity, the Behavioral Integration Model is produced by following similar activities to the ones referring to data integration, as Figure 5 shows. However, at service layer, activities focus on behavioral aspects related to the applications´ functionalities/services.

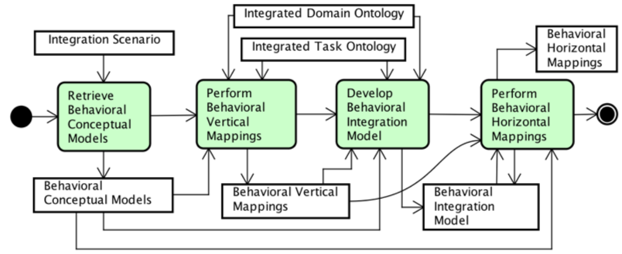

Figure 5: Integration Analysis at Service Layer

Applications' behavioral models can be described by UML activity or use case diagrams. We suggest using activity diagrams when applications to be integrated follow a workflow. In this case, the functionalities/services can be represented as a continuous flow that describes the application behavior. If it is not possible to capture the notion of workflow between the applications functionalities/services, use case diagrams are more suitable.

When performing the behavioral mappings, different from the structural mappings, which are based on the notion of semantic equivalence (e.g., the Task class in a Project Management application could be semantically equivalent to the Activity concept in the Integrated Domain Ontology), the notion used to perform the vertical mappings in the service layer is the *support relation*. Software functionalities/services are not events, as activities represent in the task ontology. Software functionalities/services are dispositions of particular computer components to do certain things [19], i.e. applications have the disposition to perform functions to support business activities. Thus, the vertical mappings between activities of the Integrated Task Ontology and the applications' functionalities/services are understood as *support relations* rather than semantic equivalences. For example, the Activity's Dates Definition service provided by a project management application could be mapped to the Define Start and End Dates to the Project Activities activity of an integrated project management task ontology, meaning that the Activity's Dates Definition service supports performing the Define Start and End Dates to the Project Activities activity.

For performing integration analysis at service level, it is necessary to consider the Structural Integration Model, since it provides the conceptual view of the data involved in the integration initiative (i.e., the Integrated Structural Model provides information related to the data manipulated by the functionalities/services). Thus, as the second step for integrating at service layer, it is necessary to perform the Harmonize Data and Service Integration activity, when data and service integration are represented together in an Integrated Service and Data Model in which the relations between services and data involved in the integration are represented (e.g., the functionalities/services parameters (inputs and outputs) are mapped to the Integrated Structural Model to ensure that the necessary data will be available in the integration solution).

5.3 Integration at Process Layer

Similar to integration at service layer (see Figure 3), integration at process layer is addressed by two activities, one (*Integrate Business Processes*) dealing with process integration itself, and other (*Integrate Services to Support Business Processes*) treating integration between process and service layers. In addition to the use of the Integrated Task Ontology as the interlingua for process mappings, BPO is used as a basis for structuring the business processes involved in the integration, so that they are defined according to the same structure and provide the information required for integration. Figure 6 details the *Integrate Business Processes* activity.

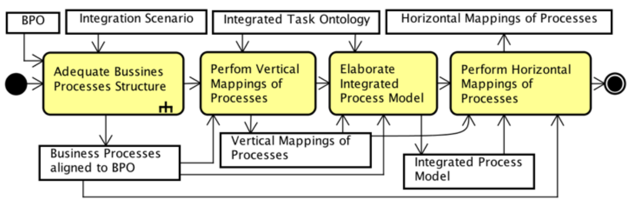

Figure 6: Integrate Business Processes activity

In *Adequate Business Processes Structure*, the business process models of the processes involved in the integration initiative should be retrieved and also aligned to the process structure defined by BPO, so that the processes are described according to a common structure that provides the necessary information for processes integration. For performing this activity, one should start by checking if there are definitions established for the business processes. If so, the definitions must be aligned to the business process structure provided by BPO. When the business process is not formally defined, it is necessary to establish a process definition aligned to BPO. To align a process definition to BPO, it is necessary to identify its sub-processes (if it is the case) and activities. For each activity, the following information should be elicited: name, description, performers, participants, inputs, results, sub-activities, pre-activities and post-activities. After aligning the process definition to BPO, the process model must be represented (e.g., using BPMN or UML activity diagrams).

It is worth noticing that the alignment of the process definitions to BPO can be an opportunity to improve the processes definition. For instance, if the processes involved in the integration initiative are defined with different levels of details, the process analyst can decide to make them homogeneous and refine the definition of some processes. Even if the process is not re-engineered, just aligning its definition to BPO can result in some improvements (e.g., BPO can help identify missing information that can compromise the process execution).

Once the business processes are properly defined and represented, it is necessary to *Perform Vertical Mappings of Processes*. The Integrated Task Ontology is used to assign semantics to the processes involved in the integration by mapping process activities to activities of the task ontology. Here, the semantic mappings represent semantic equivalence as structural mappings, i.e., process activities are mapped to activities of the task ontology with the same meaning (e.g., the Create the Project WBS (Work Breakdown Structure) process activity could be mapped to the Establish the Project Scope activity in the task ontology). Based on the vertical mappings, in the *Elaborate Integrated Process Model* activity, the Integrated Process Model representing the integrated view of the business processes involved in the integration initiative is developed. Finally, it is necessary to *Perform Horizontal Process Mappings*, identifying elements of the Integrated Process Model that have no correspondence in the Integrated Task Ontology and to map them to the applications' process models.

Integration at process layer involves not only integrating the business processes, but also integrating the services to the integrated process model, so that the process activities are supported by the services provided by the integration solution. Thus, the second activity to deal with integration at process level (see Figure 3) is Integrate Services to Support Business Processes, in which the relationship between service and process layers is established and the Integrated Process and Service Model is produced. Figure 7 details this activity.

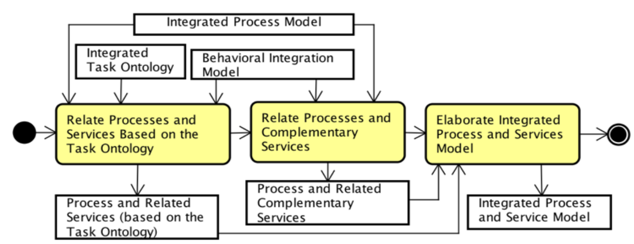

Figure 7: Perform Analysis for Processes and Services Integration activity

The first step is to Relate Processes and Services based on the Task Ontology, which consists of using the Integrated Task Ontology as a "bridge" to relate the services involved in the integration to the activities of the integrated process. As previously discussed, the Behavioral Integration Model represents the services/functionalities that should be provided by the integration solution and its support relations to activities in the Integrated Task Ontology. These mappings indicate which services/functionalities support which activities of the task ontology. The Integrated Process Model, in turn, represents an integrated view of the processes involved in the integration initiative and the mappings between the integrated process activities and activities of the task ontology. Thus, from the mappings between the services and process activities with the task ontology, it is possible to identify the relationships between the services and the activities of the integrated process. After identifying the services needed to support the process activities by using the task ontology, it is necessary to Relate Processes and Complementary Services, when services without relation with the task ontology are related to activities of the integrated process. Figure 8 illustrates the relationships between applications services and process activities. In the figure, the A1'and A2'are activities of the integrated process that map to the A1 and A2 activities of the task ontology (semantic mappings indicated by solid arrows in the figure). A1 and A2, in turn, have mappings respectively to the applications services S2 and S4 (support relations indicated by dashed arrows in the figure). Thus, it is possible to conclude that S2 is able to support A1', and S4 to support A2'. Complementarily, the S3 service, which has no mapping to the task ontology, is able to support A3'.

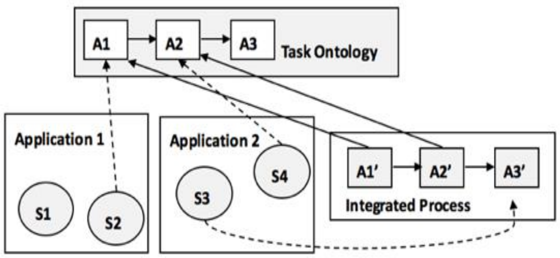

Figure 8: Process and Services relationships

Once the relations between services and process activities are identified, it is time to Elaborate the Process and Services Integrated Model, which shows how services must be combined to support the integrated process. This model can be created by using tools such as ArchiMate, which provides a modeling language for enterprise architectures that allows representing different layers, their elements and relationships between them.

6 Applying OBA--SI

After extending OBA-SI, we applied it in an AI initiative to support the Issue Management and Software Configuration Management processes. Issue Management has been increasingly recognized as a critical process for software organizations, since it addresses the identification of undesirable situations and their treatment through appropriate solutions. In order to better manage issues related to software development, we should also manage the changes made in software development artifacts and the different versions of these artifacts resulting from the implementation of solutions to issues. Thus, it is important to perform the Issue Management and the Software Configuration Management processes in an integrated manner.

The AI initiative was performed to produce an integrated solution to be used in software development projects at NEMO, the research group in which this work was carried out. The processes definitions were established by two consultancy organizations that work on software process improvement and define standard software processes to several software organizations. The integrated solution produced can be used in our research group as well as by other software organizations interested in performing those processes in an integrated way. We decided to work with processes defined by different organizations for exploring the use of BPO as a means to improve processes definition and make them more homogeneous.

By following OBA-SI, we started with *Integration Requirements Establishment* and defined the Integration Scenario shown in Table 1.

Table 1 – Integration Scenario

Goal: Improve the management and tracking of software issues, solutions, and changes resulting from these solutions.
Business Processes: Issue Management; Software Configuration Management.
Business Activities: From the Issue Management process: Register Issue, Evaluate and Prioritize Issue, Allocate Team to Issue, Identify Solution to Issue, Solve Issue, and Close Issue. From the Software Configuration Management process: Request Modification, Evaluate Request, Perform Check out, Implement Modification, and Perform Check in.
Domains: Software Issue Management and Software Configuration Management.
Applications: MantisBT (an issue tracking application) and SVN (a configuration management application).
Non-functional requirements: Use only open-source applications.

Once the integration scenario was established, we proceeded to the *Integration Analysis* phase. The first step consisted of developing the Integrated Domain and Integrated Task ontologies to be used. In this initiative, we used a fragment of SEON (Software Engineering Ontology Network) [18]. SEON is an ontology network that includes, among others, a core ontology for software processes and domain ontologies for several software engineering subdomains, such as requirements, design, coding, testing, configuration management, measurement and quality assurance. SEON specification is available at *https://nemo.inf.ufes.br/projects/seon/*. Figure 9 shows the conceptual model of the integrated task ontology used in this initiative. It provides an integrated conceptualization involving the Issue Management and the Software Configuration Management

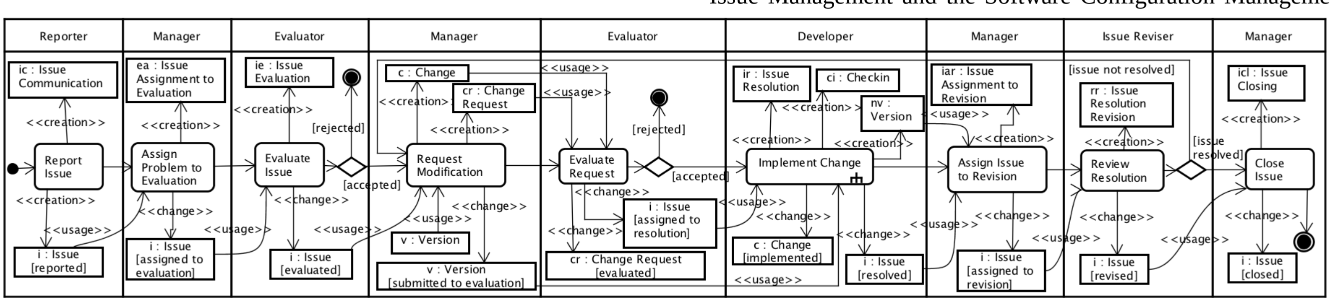

Figure 9: Integrated Task Ontology

processes. First, an issue is reported. Then, it is assigned to be evaluated. During evaluation, the issue can be accepted or rejected. issue is accepted, a modification is requested and the set of items to be modified (i.e., the version to be changed) is identified. The request is, then, evaluated and, if accepted, the change is implemented and a new version (nv) is produced. Implementing change consists in (not shown in the figure due to space limitation) performing check out of the version to be changed, modifying the version and performing check in of the new version. Once the change is implemented, the issue is assigned to revision and, if it was solved, it is closed. If the issue is accepted, a modification is requested and the set of items to be modified (i.e., the version to be changed) is identified. The request is, then, evaluated and, if accepted, the change is implemented and a new version (nv) is produced. Implementing change consists in (not shown in the figure due to space limitation) performing check out of the version to be changed, modifying the version and performing check in of the new version. Once the change is implemented, the issue is assigned to revision and, if it was solved, it is closed.

Next, we performed the integration activities related to each layer. Since this paper focuses on integration at process layer, here we discuss only aspects related to this layer. To handle integration at process layer, it is necessary to *Perform Process Integration* and, for that, we needed to *Adequate Business Processes Structure* by aligning the processes definitions to BPO. Since the processes were defined by different organizations, they had different granularity. By using BPO, we adjusted the processes definition in order to include all the necessary information (see Section 4) and make the granularity homogeneous. After aligning the processes definition to BPO, we built the processes models by using UML activities diagram. In short, the Issue Management process considered in the integration contains the following activities: Register Issue, Evaluate and Prioritize Issue, Identify Solution to Issue, Solve Issue, and Close Issue. The Configuration Management, in turn, includes: Request Modification, Evaluate Request, Perform Check out, Implement Modification, Perform Check in, Verify Modification, and Update Baseline.

Aiming to integrate these processes, we *performed Vertical Mappings of Processes*, when activities of the processes involved in the integration were mapped to activities of the Integrated Task Ontology. Table 2 presents the vertical mappings of processes. When activities are separated by "/", it means that the latter is sub-activity of the former.

After performing the vertical mappings, we *elaborated the Integrated Process Model*. Since the task ontology is to be used as interlingua, the Integrated Task Ontology model was used as a basis and the processes activities without mapping with the ontology were added to the model. The resulting integrated process contains the following activities: Report Issue, Assign Issue to Evaluation, Evaluate Issue, Identify Solution to Issue, Request Modification, Evaluate Request, Implement Change (composed of Perform Check out, Modify Version, and Perform Check in), Assign Issue to Review, Review Resolution, and Close Issue. It is possible to notice that these activities are the Integrated Task Ontology activities (see Figure 9), plus the Identify Solution to Issue activity.

Table 2 – Vertical Mappings of Processes

Integrated Task Ontology	Issue Management	Configuration Management
Report Issue	Register Issue	-
Assign Issue to Evaluation	-	-
Evaluate Issue	Evaluate and Prioritize Issue	-
Request Change	-	Request Modification
Evaluate Request	-	Evaluate Request
Implement Change/ Perform Check out	-	Perform Check out
Implement Change/ Modify Version	Solve Issue/ Implement Solution	Implement Modification
Implement Change/ Perform Check in	-	Perform Check in
Assign Issue to Review	-	-
Review Resolution	Solve Issue/ Test Implemented Solution	Verify Modification
	Close Issue/ Evaluate Implemented Solution	
Close Issue	Close Issue	Update *Baseline*

Once the Integrated Process Model was produced, we *performed Horizontal Mappings of Processes*, when activities of the integrated model without correspondence in the Integrated Task Ontology were mapped to applications' behavioral models. The behavioral models of MantisBT and Subversion were obtained during the OBA-SI activities related to integration at service layer (not detailed in this section). They represent the services provided by the applications and relevant to the integration scenario. The behavioral model of MantisBT includes the following services: Report Issue, Assign Issue to User, Update Issue and Change Issue Status. The behavioral model of Subversion, in turn, contains the following services: doCheckout, doUpdate and doCommit. The only activity of the integrated processes without a correspondence in the Integrated Task Ontology was Identify Solution to Issue. Thus, it was mapped to the Update Issue service in the MantisBT behavioral model. This service addresses, among others, the association of an issue to this solution.

After integrating the processes, we *performed Analysis for Integration of Services and Processes*. First, we used the Integrated Task Ontology as a bridge to identify the services that support the integrated process. We analyzed the vertical mappings between activities of the Integrated Task Ontology and the integrated process activities in the Integrated Process Model, and the vertical mappings between activities of the Integrated Task Ontology and applications services in the Behavioral Integration Model. Table 3 shows these last mappings, which were identified when we performed the integration activities related to the service layer. Then, we related services not mapped to the Integrated Task Ontology to the integrated process activities they support.

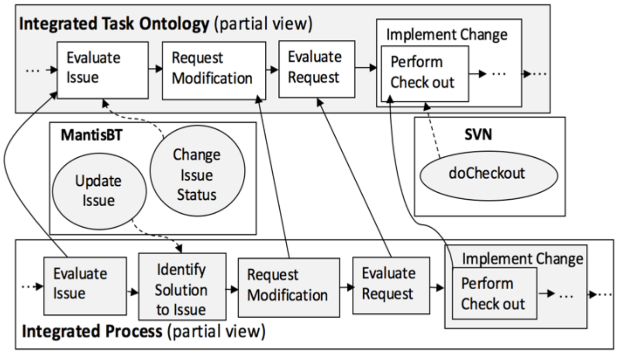

Figure 10: Relations between services and activities of the integrated process

Table 3 – Behavioral Vertical Mappings

Integrated Task Ontology Activities	Supporting Services	Application
Report Issue	Report Issue	MantisBT
Assign Issue to Evaluation	Assign Issue to User	MantisBT
Evaluate Issue	Change Issue Status	MantisBT
Implement Change/Perform Check out	doCheckout	Subversion
Implement Change/Modify Version	doUpdate	Subversion
Implement Change/Perform Check in	doCommit	Subversion
Assign Issue to Review	Assign Issue to User	MantisBT
Review Resolution	Change Issue Status	MantisBT
Close Issue	Change Issue Status	MantisBT

Figure 10 illustrates some of the identified relationships. In the figure, the Change Issue Status service supports the Evaluate Issue activity of the integrated process, since both are mapped to the same activity in the Integrated Task Ontology. Analogously, the doCheckout service supports the Perform Checkout activity of the integrated process. The Update Issue service, in turn, it is not mapped to the Integrated Task Ontology, but supports the Identify Solution to Issue activity. Based on the relationships between services and activities of the integrated process, we *Elaborated the Integrated Process and Services Model* using the ArchiMate tool. The model is shown in Figure 11. In the figure, the Process Layer illustrates the Integrated Process (process activities and the institutional roles responsible for them) to be supported by the integrated solution. The Services/Functionalities Layer shows the services provided by the applications. The relations between services and activities mean that the service is able to support the activity. Looking at the figure it is possible to notice that the applications provide services to support most of the integrated process activities. However, the Request Modification and Evaluate Request activities are not supported by any service.

Once the analysis for process and service integration was completed, it was possible to Elicit and Analyze Additional Requirements for the Integrated Solution based on the process activities not supported by services provided by the integrated applications (Request Modification and Evaluate Request in Figure 11). In order for the integrated process to be fully supported by the integrated solution, it is necessary to include new services to support these activities.

After the Integration Analysis phase, we performed activities to build the integrated solution (*design*, *implementation*, and *test*). We developed a mediator to establish communication between MantisBT and SVN and coordinate services to support the integrated process. To support the activities not covered by services provided by these applications, we added services to the mediator so that all the integrated process activities were supported by the implemented integrated solution.

7 Related work

At the beginning of this work (see Section 3), we carried out a systematic mapping of the literature [10] to investigate semantic AI initiatives (i.e., practical experiences of AI considering semantic aspects) addressing integration at process layer. Our goal was to get a panorama about the research topic and obtain useful information for extending OBA-SI. In the study, we investigated semantic AI initiatives supporting process integration in general (i.e., not only software processes). We identified and analyzed 40 initiatives. From the results, we noticed that (i) systematic approaches have not been used to guide AI at process layer; (ii) domain ontologies have been often used to assign semantics to data and service parameters, however task ontologies have not been used to support process integration; and (iii) there is a lack of a common conceptualization about processes to help process integration. Only one of the identified semantic AI initiatives focused on software processes and it used the first version of OBA-SI [3]. This is evidence that addressing semantic issues have not been a concern in AI initiatives integrating software processes.

If we disregard semantic issues, we can find several works on AI to support software processes. For example, [6] proposes an architecture to integrate tools related to the Testing process aiming at automatically generating and handling traceability links between artifacts involved in this process. In [19] it is proposed a service-oriented, metamodel-driven and process-centric AI framework to integrate applications in a tool-chain and support software processes using workflow engines. Different from OBA-SI, [6] and [19] do not propose a systematic process to guide AI. Moreover, the architecture proposed in [6] integrates only applications supporting the Testing process, while OBA-SI allows integrating applications supporting any process. Besides, as previously said, these proposals are not concerned with semantics. The new version of OBA-SI addresses semantics by using BPO and domain and task ontologies.

As for semantic AI in the Software Engineering domain, we can highlight AI initiatives carried out by using the previous version of OBA-SI [3]. In [11] two applications are integrated to support the software project management process. In [13], applications related to project management and code management were integrated to support the software measurement process. Although both the initiatives are concerned with semantics, in [11] integration covered only the data layer and in [13] integration focused on data

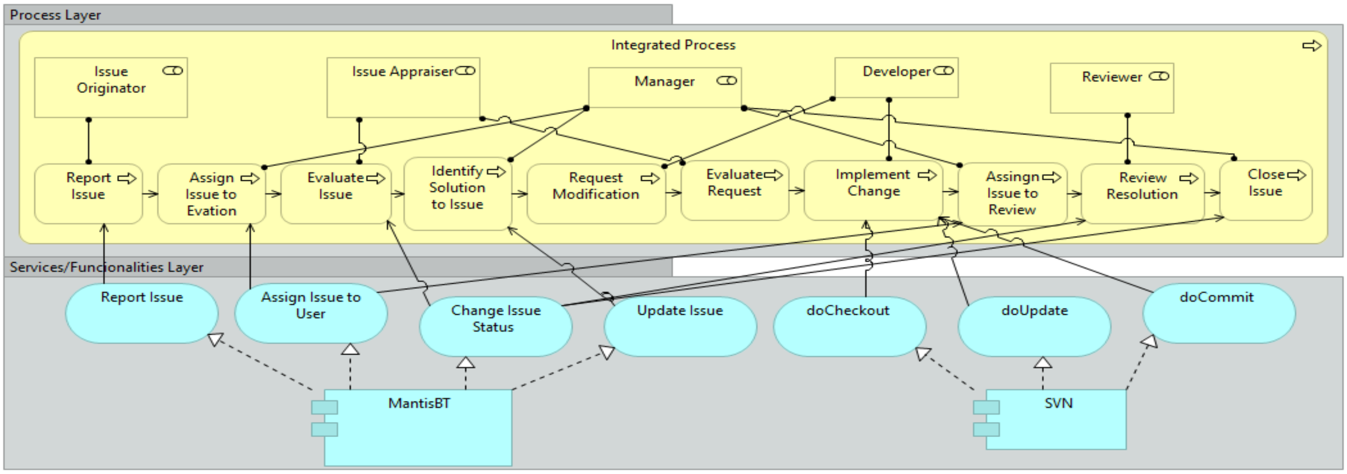

Figure 11: Integrated Process and Services Model

and service layers. None of them addressed integration at process layer.

8 Final considerations

In this paper, we presented an extension of OBA-SI [3] and its use in an integration initiative to support the Issue Management and Software Configuration Management processes. The main improvements made in OBA-SI are: (i) detailing of activities related to integration analysis at each layer; (ii) definition of the relationships between the integration layers; (iii) use of task ontologies as a basis to integration at service and process layers; and (iv) use of a Business Process Ontology to help process integration. With (i) and (ii), we made the OBA-SI integration process clearer, allowing users to understand the relations between the integration layers and address only the layers of interest in a particular AI initiative. With (iii), we improved semantics assignment at service and process layers, since task ontologies are more suitable for dealing with behavioral aspects than domain ontologies. Finally, with (iv), we provided a conceptualization about business process to be used to align and improve processes definition, and to aid in process integration.

As for the integration initiative shown in this paper, it served as a first evaluation of the new version of OBA-SI and provided initial evidence that it is feasible and the integrated solution produced by using OBA-SI is adequate, since it properly supports the integrated process. However, the initiative was carried out by the authors, thus, new evaluations are needed. As future works, we plan to use the new version of OBA-SI in other AI initiatives, involving other people, processes, and organizations. We also intend to develop a tool to support the use of OBA-SI.

ACKNOWLEDGMENTS

This research is funded by the Brazilian Research Funding Agency CNPq (461777/2014-2 and 407235/2017-5), CAPES (23038.028816/2016-41), and FAPES (69382549/2014).

REFERENCES

[1] ISO/IEC (2017) ISO/IEC12207—Systems and software engineering — Software life cycle processes.

[2] S. Izza. Integration of industrial information applications from syntactic to semantic integration approaches. Enterprise Inf.ormation Applications, Vol. 3, No. 1, Fevereiro 2009. 1-57.

[3] R.F. Calhau and R.A. Falbo. An ontology-based approach for semantic integration. In: 14th IEEE Int. Enterp. Distrib. Object Comput. Conf., IEEE, 2010: pp. 111–120. doi:10.1109/EDOC.2010.32.

[4] N. Berente, B. Vandenbosch and B. Aubert. Information flows and business process integration. Business Process Management J., v.15, 119-141, 2009.

[5] N. Guarino. Formal ontology and information systems. In: Int. Conf. on Formal Ontology in Information Systems FOIS. 1998: Vol. 46, 3-15, 1.

[6] D. Amalfitano, V.D. Simone, A. R. Fasolino and S. Scala. Improving traceability management through tool integration: an experience in the automotive domain. In: Int. Conf. on Software and System Process, 2017: pp. 5-14.

[7] O.Crelier, R.S. Silva Filho, V.M. Hasling and C.J. Budnik. Design principles for integration of model-driven quality assurance tools. In: 5th Braz. Symp. on Soft. Components, Architectures and Reuse, SBCARS, 2011: pp. 100-110.

[8] G.H. Ballinger, E.S. Fraga and R. Bañares-Alcántara. A case study in process engineering tool integration. Comp. and Chem. Eng., v.19, 1995. pp. 759-764.

[9] R. Jasper, M. Uschold. A framework for understanding and classifying ontology application. IJCAI-99, Ontology Workshop, Stockholm, 1999.

[10] L. D. Cerqueira, M. P. Barcellos, R. A. Falbo, J. C. Nardi. Process integration in semantic enterprise application integration: a systematic mapping. In: Seminar on Ontology Research in Brazil. CEUR Work. Proc. 2016. v. 1862, pp. 95-106.

[11] G. K. S. Quirino, R.A. Falbo, Integrating Tools for Supporting Software Project Time Management. In: Seminar on Ontology Research in Brazil. CEUR Workshop Proceedings. 2013. v.1041, pp.47 – 58.

[12] R.F. Calhau, R.A. Falbo, A Configuration Management Task Ontology for Semantic Integration. In: 27th ACM Symp.on Applied Computing, 2012.

[13] V.S. Fonseca, M. P. Barcellos, R.A. Falbo, An ontology-based approach for integrating tools supporting the software measurement process, Science of Computer Programming, v. 135 (15). 2017. pp. 20-44.

[14] R. A. Falbo. SABiO: systematic approach for building ontologies. In: ONTO. COM/ODISE, FOIS. 2014.

[15] M. D'aquin, A. Gangemi. Is there beauty in ontologies? Journal Applied Ontology 6 (3):165-175. 2011.

[16] G. Guizzardi. 2005, Ontological Foundations for Structural Conceptual Models. Universal Press, The Netherlands.

[17] CBOK, BPM. Guide to the business process management: body of knowledge, v. 3.0. 2013.

[18] F.B., Ruy, R.A. Falbo, M.P. Barcellos, S.D. Costa and G. Guizzardi G. SEON: a software engineering ontology network. In: 20th International Conference on Knowledge Engineering and Knowledge Management. 2016. pp 527–542.

[19] B. Polgár, I. Ágoston, G. Huhász and I. Majzik. The Integra tool integration framework. In: Symp. on Programming Languages and Software Tools. 2011. pp. 66-77