UDC 005.94:004.414.2
Original Scientific Paper

Ontology-Based Knowledge Management in Service-Oriented Systems

Ana Šaša *ana.sasa@fri.uni-lj.si*
Information Systems Laboratory,
Faculty of Computer and Information Science,
University of Ljubljana, Ljubljana, Slovenia

Marjan Krisper *marjan.krisper@fri.uni-lj.si*
Information Systems Laboratory,
Faculty of Computer and Information Science,
University of Ljubljana, Ljubljana, Slovenia

Abstract

This paper proposes an innovative framework for knowledge management in service-oriented systems. An important novelty of our framework is that it allows for automatic instance level evolution of the knowledge base and that it integrates conceptual level knowledge base evolution with the overall service-oriented system framework. In the paper, we identify several opportunities that have arisen in the field of knowledge management with the wider acceptance of SOA and its technology stack, and we present a framework that addresses these opportunities. We focus on ontology-based knowledge management systems, which have become one of the most important technologies for implementation of knowledge management systems. The framework reduces the required input for knowledge management by knowledge engineers, increases information completeness in the knowledge base and provides a simple way for storing more up-to-date information in the knowledge base.
Keywords: knowledge management, ontology, business process, service-oriented architecture

1. Introduction

This paper proposes an innovative framework for knowledge management in service-oriented systems. We have developed an architectural framework for business process oriented knowledge management in service-oriented systems and a set of business processes that integrate knowledge management with SOA (service-oriented architecture). An important novelty of our framework is that it allows for automatic instance level evolution of the knowledge base and that it integrates conceptual level knowledge base evolution with the overall service-oriented system framework. With the framework, we also strive to address several other issues in the field of knowledge management, such as integration into daily work and information gathering from different resources. We focus on ontology-based knowledge management systems, which have become one of the most important technologies for implementation of knowledge management systems [15].

In the next section, we present the background and related work in the fields of business process automation in service-oriented architecture and knowledge management. In section 3 we discuss the opportunities for knowledge management through business process automation that are the basic motivation for our research. In section 4 we present our architectural framework. In section 5, we define business processes for ontology-based knowledge management in SOA systems. Section 6 gives a discussion based on our case study. Finally, in section 7, we provide a summary and conclusions.

2. Background and related work

2.1. Business Process Automation in SOA

In the last decade, the service-oriented architecture has become the most recognized approach to business process automation. It focuses on how the information technology should support business processes of an organization with its main motivation to increase the capability of an organisation to address new business requirements on the short term by reusing existing business logic and data models, thus incurring only minimal cost, resource, and time overheads, while minimizing risks, especially when compared to rewriting entire application systems [13]. To achieve this it adds a new, service layer between the business processes layer and the application layer of an enterprise. Service layer consists of services, units of logic, which should be based on service-oriented principles, such as loose coupling, service contract, autonomy, abstraction, reusability, composability, statelessness and discoverability [6]. One of the benefits of the service layer is that composability allows further structuring of services into different layers. It is recommended that lower service layers consist of application services and that the top layer consists of process services. Process services perform the orchestration of lower level services so that their operations are executed in a certain order. In this way the execution of a business process is automated, which is an important part of achieving better agility of a business system.

Nowadays, service oriented architecture relies on web services technology. Web service orchestration requires a model with which business processes can be described in terms of connecting web services. This model is provided by BPEL (Business Process Execution Language) [16], an XML-based language designed primarily to support automated business processes based on Web services.

2.2. Knowledge management and ontologies

An important part of approaches to knowledge management is based on ontologies; some examples are [15][22][23][10][14][9]. An ontology is a shared understanding of some domain of interest [24]. It is as a means of enabling communication and knowledge sharing by capturing a shared understanding of terms that can be used both by humans and by programs [9]. Our approach is based on the OWL (Web Ontology Language) language [25] enhanced with the Semantic Web Rule language (SWRL) [26], even though it can be used with other ontology languages. OWL was chosen because of its high level of semantic expressiveness and wide acceptance of the language. OWL ontology consists of Individuals, Properties and Classes. Individuals (also known as instances) represent objects in the domain that we are interested in. Properties are binary relations between individuals - i.e. properties link two individuals together. OWL classes are interpreted as sets that contain individuals. They are defined using formal descriptions that state precisely the requirements for membership of the class. An OWL document can be used to represent two aspects of an organization: conceptual level (ontology) and instance level. In this paper, we use OWL document as a knowledge base (KB) that comprises both aspects. SWRL enables usage of implications between antecedents and consequents, where the consequent holds true if the antecedent holds [26].

Knowledge management (KM) solutions are built around some kind of an organizational memory that integrates informal, semi-formal and formal knowledge in order to facilitate its access, sharing and reuse by members of the business system for solving their individual or collective tasks [5]. The approach proposed in this paper uses the concept of OWL-based organizational knowledge base (OOKB). The concept of organizational knowledge base, sometimes also referred to as corporate memory or organisational memory, refers to stored information from an organization's history that can be brought to bear on present decisions [27] and supports sharing and reuse of individual and corporate knowledge and lessons learned [1]. Considerable research effort has been dedicated to this field of science; examples are [27][1][2][5][10]. Several researchers have been concerned with application of organizational memory approaches to the business process execution domain [1][18][11];

however, we have observed that most of these approaches were developed more than a decade ago, and that in the last years the advances in IT have brought several opportunities that could significantly improve some of the major concerns that exist in this field. In this paper we discuss these opportunities and propose a framework that integrates the recent development in the domain of business process automation and organizational memory systems.

3. Knowledge Management through Business Process Automation

In this chapter, we present the motivation for our research. An important characteristic of our approach is that we are not trying to propose a completely new approach to knowledge management or knowledge management systems development. Instead, the proposed approach can be applied to the chosen (existing) approach to KM in order to enhance it with the opportunities which have arisen with the advent of service-oriented business process automation.

Knowledge management facilitates the capture, storage and dissemination of knowledge using information technology [14]. Fischer and Ostwald [8] view knowledge management as a continual process which 1) creates knowledge as a by-product of work, 2) integrates the knowledge in an open and evolving repository and 3) disseminates it to others in the organisation when it is relevant to their work. The proposed framework is built on these observations; within the business process context they translate into the following: knowledge management is a continual process which 1) creates knowledge as a by-product of business process execution, 2) integrates knowledge in an open and evolving repository and 3) disseminates it to others in the organisation when the process activities require it.

With the spread of SOA and SOA-related technologies, we have observed several opportunities which we see as important potential contributors to KM systems. These opportunities are:

a) Automatic translations of process data to OOKB:

All the business activity for a given business system can be described with its business processes. Thus, every information component relevant and known to the business system is processed (e.g. created, retrieved, used or changed) in at least one of the business processes. Let us suppose that an OOKB makes part of the information system and that the information system automates one or more business processes with respect to the principles of service-orientation. In the context of process activities, the process data have a certain meaning and should therefore correspond to information about instances in the OOKB. Therefore, a translation between the process data and instance level of the OOKB exists. In a SOA system, instance level data can be captured through business process execution if such translations are defined.

One of the important visions of SOA is to achieve automation of a large part of business processes in a way that enables flexible, cost and time effective introduction of changes. In a fully implemented SOA system, execution of all the business processes would be automated by a business process execution engine orchestrating different services. In such system, every instance in the OOKB would have a corresponding source in a business process and could be automatically translated into the OOKB. Even though we usually do not deal with fully implemented SOA systems, translations in the implemented business processes would represent an important advantage to the support of instance level changes to the ontology.

Data are a basis for information. They represent facts, symbols that describe some content, without a wider context. Information represents processed data that have a certain added meaning for the user [12]. In the business process execution context, process data are the basis for input information required to accomplish process activities. In order to use them together with the information in the OOKB and to enable reasoning upon them we have to place them into correct context. This context is determined by the business process model of a business system (business process context). Furthermore, in different process and activity instances, different data can represent different information depending on the actor that performs an activity (or is responsible for it in case of automated execution). This is particularly important in case of human tasks. Therefore, the context also depends on the entity (actor) responsible for the activity, for example an individual, a department, a business system as a whole or other business entity.

Process data correspond to the instance level of the OOKB. To place process data into correct context we define translations to the OOKB. The goal is to enable automatic capture of information about instances of the OOKB during business process execution. Review and confirmation may be required. In this case the translation is semi-automatic

Structure of process data is defined by XML schemas. The basis for defining a translation is thus the XML schema of the process data and the ontology level of the OOKB. All process data is not always relevant for execution of a process activity. We refer to the process data that represent the input of the translation as process information carriers. A process information carrier can be any part of the process data. It can be represented by a single XML element, an occurrence of an element within another element, a combination of several elements etc.

b) Gathering information from different sources with the SOA approach to integration:

Service-oriented integration utilizes solutions based on interoperable services. Because of the vendor-neutral communications framework established by web services-driven SOA, the potential is there for enterprises to implement highly standardized service description and message structures [6]. Gathering information from different sources in the SOA can be therefore realized regarding the principles of service orientation and using the same integration platform. For example, let us presume that the information systems which contain information required for an activity expose a web service with an operation which returns the required information. In this case, information gathering could be implemented with a BPEL process invoking the corresponding services. The required information would be obtained through the service output.

Thus, the SOA approach to integration can be also used for gathering information from different sources, instead of gathering information from different sources in a way that leads to tight coupling with the KM system used. For this, the OOKB itself can be exposed as a web service. In this way, we can enable more flexibility, when the source information systems change through time, and independence of the specific KM system used. The proposed framework is therefore based on principles of service orientation and is composed of several loosely coupled services that comply with these principles.

c) Automatic queries of the OWL knowledge base:

In a SOA system, business processes are implemented as BPEL processes orchestrating different services required for process execution. For this, exact business process models and process data are required to be defined. In case of activities of a highly unpredictable nature, all the data components would be difficult to define in advance. However, in many cases, the activities of a business process for which the OOKB is required along with the required data components can be identified. For these activities, the required query of the OOKB can be specified (statically or dynamically). At runtime they can be automatically executed.

d) Integration into daily work:

As discussed in c), business process formalization enables that a significant part of the activities that require the OOKB can be identified along with the required information component. This is especially useful in case of activities that are performed by human actors (human tasks). When defining a human task, we can often identify what data components will be needed for its execution. Therefore, OOKB queries can make part of a human task definition. Before the task instance is presented to the person who performs the task (task owner) the query can be executed. The information obtained can be represented to the task owner with the task. In this way, for the task owner the use of the OOKB is transparent.

When ontology evolution activities require changes to the ontology, the responsible knowledge engineer has to implement these changes in the ontology. These activities are a part of the ontology evolution business process. As they require human work, they can be implemented as human tasks. When a change to the OOKB needs to be implemented, a human task is triggered and assigned to the responsible knowledge engineer. The events that trigger these activities and how they should be performed are defined with the chosen KM methodology. Approach proposed in this paper does not define a specific KM methodology and can be used with an arbitrary KM methodology. In our framework ontology development and evolution is XML schema management driven.

e) Automation of certain knowledge-based human performed tasks:

If knowledge about activities is captured in a way that can be automatically processed and if the required activity inputs are available, the task can be automatically executed. Activities that require a reasoning process to be performed are rarely automated because they require human reasoning and logical abilities. However, often reasoning activities require reasoning that is of a repetitive nature – different activity instances require the same or similar reasoning process. In business systems, human task owners are also bounded by the business system's policies and rules that they have to respect. For the repetitive types of reasoning tasks techniques exist that can be used in order to capture this type of knowledge, for example decision rules, business rules, logical statements etc. Based on a) and b), the required input for reasoning activities can be retrieved from the corresponding source if it is available. Therefore, if the knowledge required for reasoning is captured in the OOKB, a reasoning task can be automated by executing a query and reasoning with the OOKB. This means that with appropriate planning, choosing the appropriate techniques and applying them to the ontology-based KM system as part of the overall SOA system, a higher level of business process automation can be reached. The authors have discussed such automation in more detail in their previous work [20].

4. Architectural Framework

In this section we represent the architectural framework for ontology-based knowledge management in service-oriented systems. Based on the observations discussed in the previous chapter, the architectural framework is designed based on the principles of service-orientation in a way that supports automatic translations of process data into OOKB (section 3a), allows for gathering information from different resources using the SOA integration platform (section 3b), enables automatic queries of the OWL knowledge base (section 3c), can integrate knowledge management into daily work (section 3d), and allows for automation of knowledge-based human tasks (section 3e).

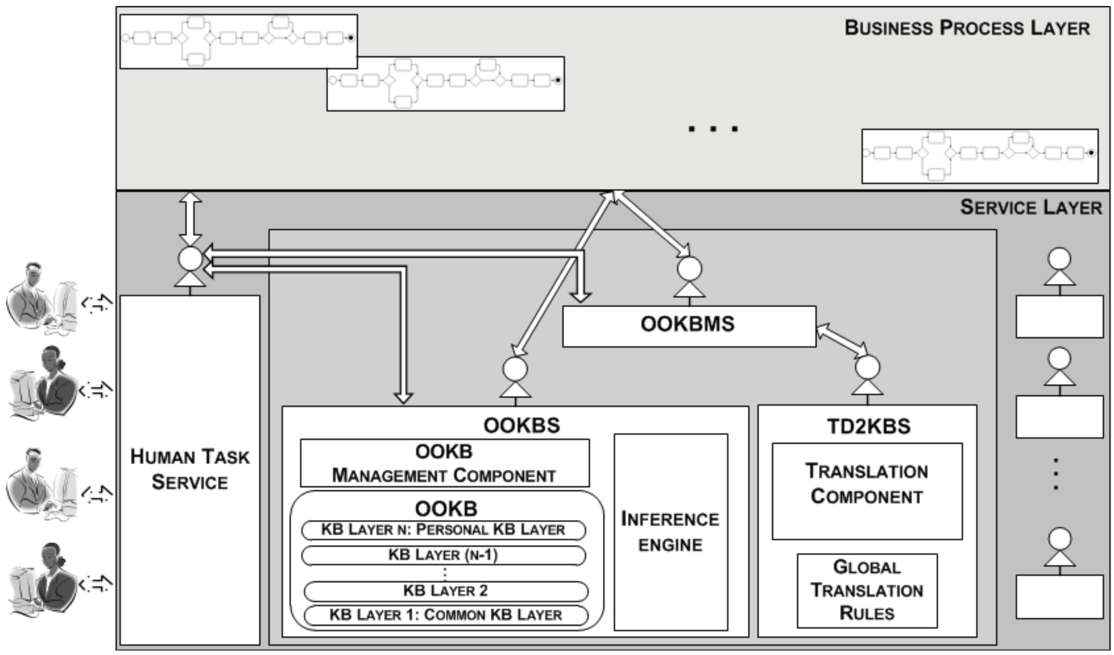

Figure 1. Architectural Framework Overview

By applying the principle of service orientation to these desired functionalities, we have developed a framework composed of three services (Figure 1): 1) OOKB Service (OOKBS), which is concerned with OOKB issues, such as consistency, validation, reasoning, evolution etc., 2) Translation of Data to KB Service (TD2KBS), which is concerned with translation of data to the concepts of the ontology, and 3) OOKB Management Service (OOKBMS), which is a composite service concerned with correct management of requests for ontology activities from the business process layer and is responsible for correct orchestration of the two other services. The OWL-based organizational knowledge base is the key part of the OOKBS.

Figure 1 also illustrates our framework in the scope of the overall SOA environment. The business process layer illustrates the use of the OOKBMS in business processes. The Human Task Service is the service that supports tasks of the business process that require human actions. The Human Task Service is a common component of SOA systems, and we show its relation to our framework. Figure 1 demonstrates that Human Task Service is used in business processes as well as by the OOKBS. The OOKBS uses the Human Task Service when review and confirmation of changes is required, for example for the critical concepts, or when a need for conceptual level changes is discovered. The Human Task Service can use the OOKBMS in order to integrate automatic queries into human task execution and to enable automatic human task execution, if possible.

In the remainder of the section, we first describe the OWL-based organizational knowledge base (OOKB) and then we represent the main operations for each of the services in our framework.

4.1. OWL-based organizational knowledge base

The basic structure of an OOKB is illustrated in Fig. 1. Hierarchical structure is proposed due to some important advantages, such as reducing search complexity and promoting reuse of stored knowledge [14]. Let us say that the term *organisational structure element* (OSE) is a generic term for any kind of element in the organisational structure; it may be an organisation itself, a department, a role, a group or any other possible structural element used in an organisation.

OOKB comprises several OWL KBs, each belonging to a different OSE. Every OWL KB of the overall OOKB belongs to exactly one OSE. If OSE_1 is a part of OSE_2, OSE_1 can use the OSE_2's OWL KB. The common KB layer represents the organisation's common knowledge, such as its organisational structure and common business policies. The personal KB layer comprises personal KBs. They contain knowledge concerning people and their roles in the business system.

This paper does not propose a new methodology for ontology or KB development or evolution. For this purpose, an arbitrary existing methodology can be chosen and used, such as [15][22][23]. Our approach is complementary to these methodologies.

4.2. OOKB Service

The OOKBS service has three main operations:

- *Insert*:
  Input of the operation is composed of OWL data and an OSE. Based on the OSE, the OWL data provided as the input is added into the OSE's KB. This may result in an exception if it causes inconsistencies in the OSE's KB. In this case the new information is discarded and not stored in the OSE's KB.
- *InsertAndRetrieve*:
  Operation is the same as the Insert operation, except that the updated OSE's KB is also returned with the output.
- *Query*:
  Input of the operation is composed of a list of queries and an OSE. Reasoning is performed on the OSE's KB and the queries are performed upon the inferred KB. The output comprises query results.

- *QueryMerged*:
  Input of the operation is composed of the OWL data, an OSE and a list of queries. Based on the OSE, the OWL data provided as the input is merged with the OSE's KB and returned as the output. The OSE's KB does not change with this operation, but is only retrieved and merged with the provided OWL document. Reasoning is performed on the merged OWL document and the queries are performed based on the result of this reasoning. The merge may result in an exception, for example if inconsistencies are detected in the merged OWL document.
- *QueryOWLInput*:
  Input of the operation is composed of an OWL document and a list of queries. Reasoning is performed on the OWL document and the queries are performed based on the result of the reasoning. The output is composed of the query results.

4.3. Transformation of Data to Knowledge Base Service

The TD2KBS service has one operation:

- *TranslateD2O*:
  Input of the operation is composed of process data and translation rules. Translation rules are an optional input. The TD2KBS translates the process data into the OWL concepts based on the translation rules. If they are not provided with the input, the global translation rules are used. Output of the operation is an OWL document obtained from the process data. If the translation is not successful an exception is thrown. If review and confirmation is required for the translation, a human task is invoked for the responsible knowledge engineer who revises the translation.

4.4. OOKB Management Service

The OOKBMS service has three main operations:

- *QueryWithStore*:
  Input of the operation is composed of process output queries, input process data, translation rules and an OSE. Input process data and OSE are optional inputs. If input process data and OSE are provided, it is translated to OWL concepts by TD2KBS and merged with the OSE's KB by the InsertAndRetrieve OOKBS operation. Afterwards the Query operation of the OOKBS is invoked with the list of output queries and the OSE. Result of the queries is provided as the output of the QueryWithStore operation.
  If OSE is provided with the input without the input process data, the Query operation of the OOKBS is invoked with the list of output queries and the OSE. Result of the queries is provided as the output of the QueryWithStore operation.
  If input process data is provided with the input without the OSE, then the input process data is translated to OWL by the TD2KBS operation. The QueryOWLInput operation is invoked with the translated OWL document and the list of queries. Result of the queries is provided as the output of the QueryWithStore operation.
  If OOKBS or TD2KBS return exceptions, an exception is also thrown by the QueryWithStore operation.
- *QueryWithDiscard*:
  The operation is the same as the QueryWithStore operation in all aspects, except that if the process data input and OSE are provided the QueryMerged operation is called instead of the InsertAndRetrieve operation.
- *AddProcessInformation*:
  Input of the operation is composed of process data and OSE. Both are required to be provided. First, the TranslateD2O operation is invoked to obtain OWL representation from the process data, and then the Insert operation is invoked with the resulting OWL document and the OSE to add instance information to the OSE's KB.
  If OOKBS or TD2OS return exceptions, an exception is also thrown by the AddProcessInformation operation.

Other exceptions may be thrown by the operations, for example due to faulty input or execution issues.

5. Business processes for ontology-based knowledge management in SOA systems

This section discusses how OOKB development and evolution of the OWL-based organizational knowledge base is supported and integrated with the proposed architectural framework and business processes of an organization. In our framework the OOKB evolves based on the overall SOA architecture and business processes. In order to define and maintain the translations and to ensure that the OOKB encompasses relevant and up-to-date information, we have defined three business processes that incorporate definition of translations and evolution of OOKB ontology into the overall scope of business process oriented SOA. All the three business processes are integrated into SOA development and maintenance processes of the organization. They are illustrated in Fig. 2-4 in the Business Process Model and Notation 2.0 [17]. In the following subsections, each of these processes is described in more detail.

5.1. Creation of new translation and OOKB points business process

In our framework, the process of creation of a new translation to OOKB depends on the overall XML schema development. BPEL processes use XML data based on XML schemas, which should comply to SOA governance regulations used in the organization [3]. When a new XML schema is developed or an existing schema is updated with new element(s), a knowledge engineer should determine whether the data that will be based on the new element(s) should be stored in the OOKB or not. If it should be, then the corresponding translations have to be defined.

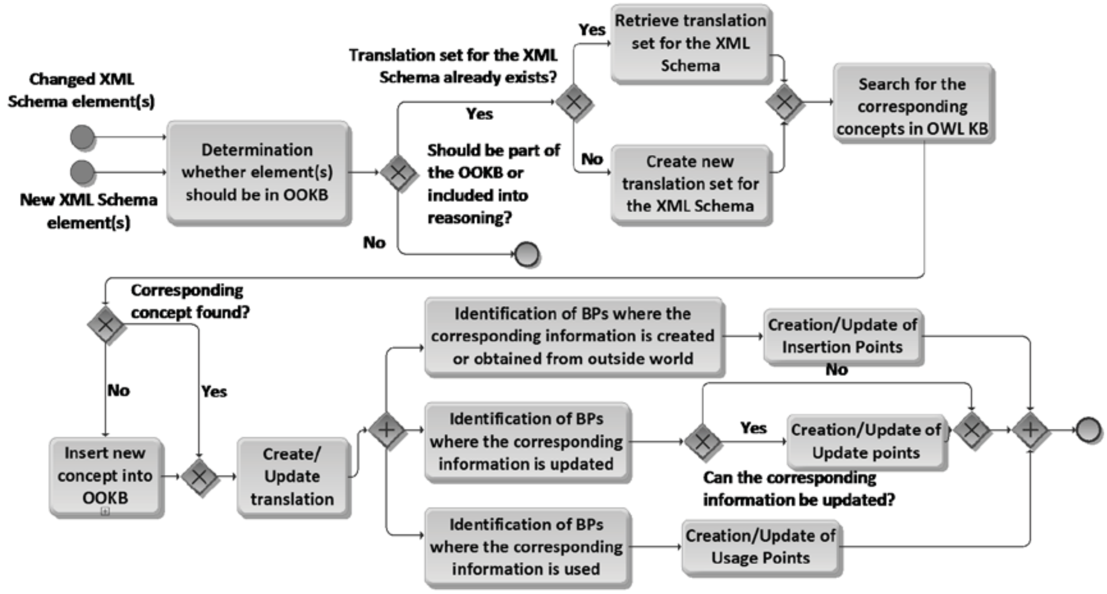

Figure 2. Creation of new translation and OOKB points business process

Our framework also provides support for cases, for which storage is not required or desired, however reasoning capabilities on the new data can contribute to the business process. Therefore, the corresponding translations are defined also in these cases.

As every schema can have a translation for each of its elements, and since there can be several translations for one XML element (for different contexts and OSEs for example), every schema can have a corresponding translation set. For every XML element that will have a representation in the OOKB, the corresponding concept has to be identified in the OOKB by the knowledge engineer. If such a concept is not found, this means that it has to be created in the KB. These activities are comprised in the Insert new concept into OOKB subprocess. Insertion of a new concept into the OOKB should comply to the chosen ontology evolution methodology. The subprocess is thus determined by the chosen methodology.

Afterwards, a translation is defined between the XML schema element and the OOKB. In case of an updated XML element, an existing translation can be only updated. After defining a translation, business processes where the data based on the schema is created or obtained from the organization's outside world need to be determined. Similarly, business processes where this data is used have to be identified. If there is no business process that used the data, then the data is not necessary and should not be stored. If the data can be updated after its creation/retrieval from the outside world, then the business processes which update it, need to be determined. After identification of the corresponding business processes, the exact points where the creation/usage/update occurs in the business process flow need to be identified. We refer to these points in business processes as the OOKB points and we distinguish the three types based on whether the data is new (insertion points), updated (update points) or retrieved (usage points). A special type of a usage point is when the data is not stored in the OOKB but only used for reasoning:

- If it is used for reasoning with existing OOKB, then we refer to it as light-merge usage point.
- If it is used for reasoning only based independent of OOKB, then we refer to is as a light usage point.

OOKB points have to be identified and created. In a BPEL process, the OOKB points represent invocation of a corresponding OOKBS operation:

- Insertion point: the Insert operation is invoked.
- Update point: the Insert operation is invoked.
- Usage point: the Query operation is invoked.
- Insertion or Update point followed by an Usage point: InsertAndRetrieve operation is invoked.
- Light merge usage point: the QueryMerged operation is invoked.
- Light usage point: the QueryOWLInput operation is invoked.

In case of an updated XML element, existing OOKB points can also be updated.

5.2. New or updated BPEL process-based OOKB maintenance business processes

Creation of new translation and OOKB points for a certain information carrier type have been defined, new OOKB points may be developed or existing OOKB points may change with new BPEL processes or changes of the existing BPEL processes. The processes defined in this section concern these issues.

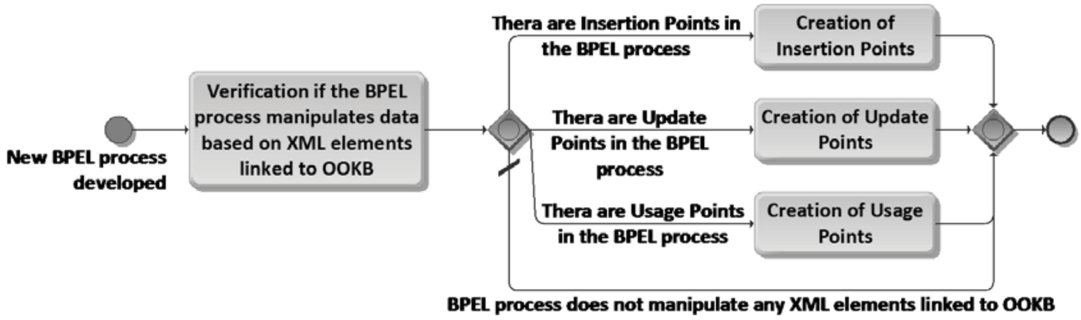

Figure 3. New BPEL process-based OOKB maintenance business process

When a new BPEL process is developed (Figure 3), a knowledge engineer should finalize it with the OOKB points in order to integrate it into the overall OOKB framework. The knowledge engineer should determine whether the process operates with any XML data that is linked to the OOKB, either by creating information of interest to the OOKB, either by updating it or by using it. If it is linked to OOKB, the OOKB points need to be identified and created.

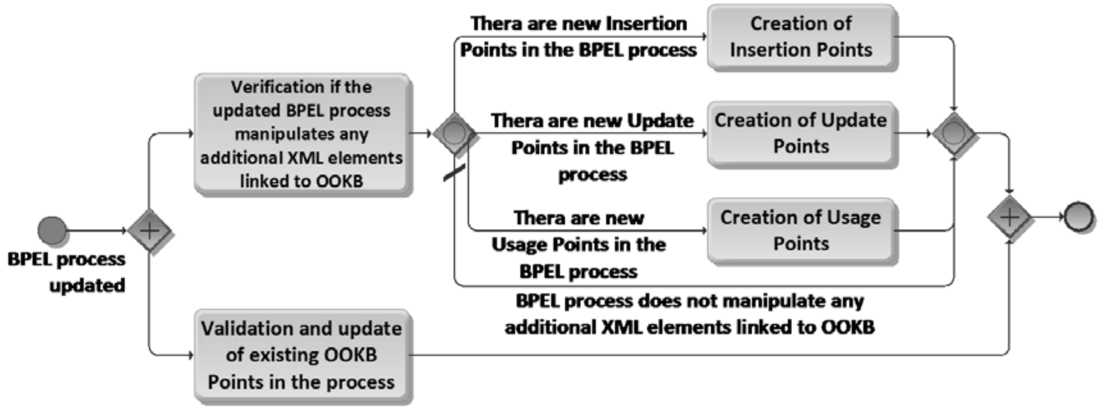

Figure 4. Updated BPEL process-based OOKB maintenance business process

In a similar manner, when a BPEL process is updated, a knowledge engineer should finalize the update by verification and validation of the existing OOKB points and adding new OOKB points if needed (Figure 4).

6. Discussion

We have performed a case study for the framework for the product resale domain. As the SOA environment, we have used the Oracle SOA Suite 11g. Using this environment, we have implemented the OOKBMS, TD2KBS, and OOKBS services. For the translation definitions and implementation of the run-time translations, we have adopted and extended the JXML2OWL translation approach [19]. There are other potential approaches that can be used, such as [7] or [4]. An example XML element defining a translation from input data to the ontology is represented in Listing 1.

```
<os:ontologyTranslationElements >
    <os:businessEntity>
        Human Resources Department
    </os:businessEntity >

    <os:translateToOWLIndividual refName="translationToCandidates">
     <os:elementForOWLIndividual type="xsd:string">
        $candidate
     </os:elementForOWLIndividual>
     <os:IDForOWLIndividual type="xsd:string">
        $candidate/personalDetailes/ID
     </os:IDForOWLIndividual>
     <os:OWLClass">
        http://www.oo.si/candidates.owl#Candidate
     </os:OWLClass>
  </os:translateToOWLIndividual>

  <os:translateToOWLDatatypeProperty>
  <os:referenceToTranslationToIndividual name=”translationToCandidates”>
     <os:OWLDatatypeProperty>
        http://www.oo.si/ candidates.owl#yearsOfWorkExperience
     </os:OWLDatatypeProperty>
     <os:datatypeValue type="xsd:int">
        fn:sum(for $j in $candidate/workExperience return $j/:nbOfMonths)
     </os:datatypeValue>
  </os:translateToOWLDatatypeProperty>

  <os:translateToOWLObjectProperty>
     <os:referenceToTranslationFromIndividual name=”translationToCandidates”>
```

```
    <os:toIDOfExistingOWLIndividualFromInput type="xsd:string">
          for $j in $candidate/foreignLanguages/foreignLanguage
          return
            if(fn:compare($j/code,'en')=0) then
              fn:concat('http://www.oo.si/candidates.owl#',fn:data($j/p1:level))
            else ('http://www.oo.si/candidates.owl#Level_0')
    </os:toIDOfExistingOWLIndividualFromInput>
    <os:OWLObjectProperty>
       http://www.oo.si/candidates.owl#levelOfEnglishLanguage
    </os:OWLObjectProperty>
  </os:translateToOWLObjectProperty>

</os:ontologyTranslationElements>
```

Listing 1: Example translation rules

In order to support automatic querying and query answering based on the information in the OOKB and on the information obtained by reasoning upon the OOKB, the input of the OOKBMS needs to specify the desired output of the query. This specification can be seen as a question that we are trying to answer. An example of a query language that can be used with an OWL ontology is SQWRL (Semantic Query-Enhanced Web Rule Language). An example question "Who is the best candidate for the position X?" in SQWRL could be written as follows:

$$\text{Person}(?\,x) \wedge \text{isBestCandidateForPosition}(?\,x, ?\,p) \rightarrow \text{sqwrl: select}(?\,x)$$

When an OOKBMS service requester wishes to query the OOKB, a query needs to be formed. In BPEL business processes, a query is formed at run-time based on process data using the XPath language. It is passed to the OOKBMS through a usage point. The OOKBMS queries the OOKB and returns the result of the query back to the process instance. Listing 2 represents an example of a query for an output element candidate with two subelements – candidateName and candidateID. A query is divided in two parts and every select statement defines the value of a subelement. When query is executed, the query definition is replaced by the values it returns.

```
<os:candidate>
    <os:candidateName type="xsd:string">
            Person(? x) ∧ isBestCandidateForPosition(? x, ? p) ∧ hasID(? x, ? id) → sqwrl: select(? x)
    </os:candidateName>
    <os:candidateID type="xsd:string">
         ∧ sqwrl: select(? id);
   </os:candidateName>
</os:candidate>
```

Listing 2: Output query example

In our case study, we have developed an OOKB with the main concepts and instances of the domain. All the subsequent evolution of the OOKB was strictly XML schema driven. This means that new concepts of the OWL ontology were added based on new XML elements in XML schema. Other types of events did not influence the OOKB. Our framework presumes this behavior based on SOA principles and the observation that if a new concept is required, it means that it occurs in at least one of the business processes of the organization. In SOA-driven development of business processes, the concept is supported by an XML schema element, and therefore a new XML schema element is an appropriate event to trigger evolution activities of the OOKB. In our case study, this approach resulted in development of a valuable KB that was used in the BPEL processes.

In our case study, we found our approach to be very useful to integrate the ontology-based knowledge management into the overall scope of SOA-drive business processes. We found the framework to be a simple but a powerful tool to enable a holistic OOKB creation, evolution, management and usage. In order to successfully use our approach, special attention should be paid to structuring and handling translations and translation sets. With increasing number of BPEL processes and XML schema elements, the number of translation sets increase as well as the number of all the translations. For implementation of our approach, it is therefore important that the XML schemas are globally managed at the organization level and that the translation sets are always related to a schema. Since the instance level of the OOKB depends entirely in the translations, the increased number of translations would make it very difficult to maintain the OOKB up-to-date and in a consistent state, if every translation sets was not clearly tied to an XML schema. A global repository of translation sets, translations and their attributes, such as the corresponding OSE and level of granularity, is a useful mechanism that can improve the OOKB governance.

7. Conclusion

In this paper we have presented a novel framework for integration of knowledge management and SOA systems. We have identified several opportunities that have arisen in the field of knowledge management with the wider acceptance of SOA and its technology stack: automatic translation of process data into organization's knowledge base, integration of knowledge management activities into daily work, using the SOA platform for information gathering from different resources, automatic knowledge base querying from business processes, and automation of certain knowledge-based human tasks. We have developed an architectural framework that addresses these opportunities and a set of business processes that integrate knowledge management using the proposed architectural framework. These processes are: Creation of new translation and OOKB points, New BPEL process-based OOKB maintenance, Updated BPEL process-based OOKB maintenance business process.

Our case study has shown that our framework is a very useful for integration of ontology-based knowledge management with the overall scope of SOA-drive business processes. It is a simple but a powerful tool that enables holistic organizational knowledge base creation, evolution, management and usage. Furthermore, it reduces the required input for knowledge management by knowledge engineers, reuses process data for instance-level knowledge base, and consequently increases information completeness in the knowledge base by systematic identification of organizational knowledge base points in the business processes of the organization and by taking into consideration more up-to-date information inserted at run-time through the executing business processes.

References

[1] Abecker, A.; Bernardi, A., Hinkelmann, K.; Kühn, O.; Sintek, M. Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3): 40–48, 1998.

[2] Anand, V.; Manz, C.C.; Glick, W.H. An Organizational Memory Approach to Information Management. *The Academy of Management Review*, 23(4): 796-809, 1998.

[3] Biske, T. *SOA Governance*, Packt Publishing, 2008.

[4] Bohring, H.; Auer, S. Mapping XML to OWL Ontologies. In *Proceedings of 13. Leipziger Informatik-Tage*, 2005.

[5] Dieng, R.; Corby, O.; Giboin, A.; Ribière, M. Methods and tools for corporate knowledge management. *International Journal of Human Computer Studies*, 51(3): 567-598, 1999.

[6] Erl, T. Service-Oriented Architecture: Concepts, Technology and Design. Prentice Hall, Upper Saddle River, NJ, 2005.

[7] Ferdinand, M.; Zirpins, C.; Trastour, D. Lifting XML Schema to OWL. In Proceedings of Web Engineering - 4th International Conference, pages 354-358, Munich, Germany, 2004.

[8] Fischer, G.; Ostwald, J. Knowledge Management: Problems, Promises, Realities, and Challenges. *IEEE Intelligent Systems*, 16(1):60 – 72, 2001.

[9] Gruber, T.R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human Computer Studies*, 43(5-6): 907-928, 1995.

[10] Huang, C.-C., Tseng, T.-L., Kusiak, A. XML-Based Modeling of Corporate Memory. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(5): 629 – 640, 2005.

[11] Kaathoven, R. van; Jeusfeld, M. A.; Staudt, M.; Reimer, U. Organizational Memory Supported Workflow Management. In *Proceedings of the 4. Int. Tagung Wirtschaftsinformatik*, pages 543-563, Heidelberg, Germany, 1999.

[12] Kavčič, B. Poslovno komuniciranje, Celje: Visoka komercialna šola, Ljubljana, 2006.

[13] Krafzig, D.; Banke, K.; Slama, D. *Enterprise SOA: Service-Oriented Architecture Best Practices*, Prentice Hall PTR, 2004.

[14] Lai, L. F. A knowledge engineering approach to knowledge management. *Information Sciences*, 177(19): 4072-4094, 2007.

[15] Maedche, A.; Motik, B.; Stojanovic, L.; Studer, R.; Volz R. Ontologies for Enterprise Knowledge Management. IEEE Intelligent Systems, 18(2): 26-33, 2003.

[16] OASIS. Web Services Business Process Execution Language Version 2.0, OASIS Standard. 2007. Available on: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

[17] OMG, Business Process Model and Notation (BPMN), Version 2.0. 2011. Available on: http://www.omg.org/spec/BPMN/2.0/.

[18] Reimer, U.; Magelish, A.; Staudt, M. EULE: A Knowledge-Based System to Support Business Processes. *Knowledge-Based Systems*, 13(59): 235-331, 2000.

[19] Rodrigues, T.; Rosa, P.; Cardoso, J. Moving from syntactic to semantic organizations using JXML2OWL. Computers in Industry, 59(8): 808-819, 2008.

[20] Šaša, A.; Jurič, M. B.; Krisper, M. Service-Oriented Framework for Human Task Support and Automation. IEEE Transactions on Industrial Informatics, 4(4): 292-302, 2008.

[21] Schreiber, G.; Akkermans, H.; Anjewierden, A.; Hoog, R. de; Shadbolt, N.; Velde, W. Van de; Wielinga, B. *Knowledge Engineering and Management: The CommonKADS Methodology*, The MIT Press, 1999.

[22] Staab, S.; Schnurr, H.-P.; Studer, R.; Sure, Y. Knowledge processes and ontologies. IEEE Intelligent Systems, 16(1): 26-34, 2001.

[23] Sure, Y., Staab, S., Studer, R.: Methodology for development and employment of ontology based knowledge management applications. SIGMOD Record-Web Edition, 31(4), 2002.

[24] Uschold, M.; Gruninger, M. Ontologies: Principles, methods and applications. Knowledge Engineering Review, 11(2):93-136, 1996.

[25] W3C. OWL 2 Web Ontology Language, Document Overview. 2009. Available on: http://www.w3.org/TR/owl2-overview/.

[26] W3C. SWRL: A Semantic Web Rule Language, Combining OWL and RuleML. 2004 Available on: http://www.w3.org/Submission/SWRL/.

[27] Walsh, J.P.; Ungson, G.R. Organizational Memory. *The Academy of Management Review,* 16(1): 57-91, 1991.