Towards Ontology-Based Methodology for Requirements Formalization

Eduard Babkin and Ekaterina Potapova

State University – Higher School of Economics,
603155, B. Pecherskaya, 25/12, Nizhny Novgorod, Russian Federation
eopotapova@mail.ru

Abstract. Implementation of enterprise information management systems is still a challenging task for any organization. One of the key challenges within implementation projects is analysis of business requirements and determination of required system capabilities. Traditionally this challenge is overcome by gathering a team of experienced specialists but we would like to propose an alternate solution: using the ontology based knowledge management system to determine the necessary functionality and configuration of the enterprise information management system based on the user requirements. In this paper we describe the approach for representation of user requirements for such systems as business processes based on the ideas from The Ontology for Linking Processes and IT infrastructure (OLPIT).

1 Introduction

Requirements engineering has always been an actual aspect of software development processes. Depending on the technology and application domain the approaches for requirements analysis may differ but in any situation the main objective of the requirements engineering process is to provide a model of what is needed in a clear, consistent, precise and unambiguous statement of the problem to be solved.

Statistics shows the importance of proper requirements engineering approach for the software development project. Studies by Boehm [1, 2] and others have shown that the potential impact of poorly formulated requirements is substantial. Boehm suggested that requirements, specification and design errors are the most numerous in a system, averaging 64% compared to 36% for coding errors. Most of these errors are not found during the development stage but at the testing and delivery stages. The resulting cost to correct these bugs increases with the time lag in finding them. A requirements error found at the requirements stage costs only about one-fifth of what it would if found at the testing stage, and one-fifteenth of what it would cost after the system is in use.

The criticality of requirements engineering stage should be also underlined for projects that are related to the implementation of Enterprise Resource Planning (ERP) systems. Here ERP means not only the systems that functionality corresponds to ERP standards, but also any corporate information management system that helps to manage key areas of the business from the beginning to end, e.g. customer relationship

P. Forbrig and H. Günther (Eds.): BIR 2010, LNBIP 64, pp. 73–85, 2010.

management systems, supply chain management systems, strategic planning systems, etc. Surveys have shown that inadequate definition of functional requirements accounts for nearly 60% of ERP implementation failures. This is simply a matter of not comprehensively and systematically developing a quality set of functional requirements definitions which lead to the misfit of application software with business processes, miscalculation of time and effort and inadequate training and education for the end users.

Currently most of ERP vendors propose to use 'off-the-shelf' (OTS) [10, 5] methodology for systems implementation. Within OTS approach the requirement engineering phase is built on the fit-gap analysis technique. Essentially, such an off-the-shelf process is composition and reconciliation: the logic behind it is to start with a general set of business process and data requirements and then explore standard ERP functionality to see how closely it matches the company's process and data needs. Practical use of fit-gap analysis technique has shown a set of problems that are often met by project teams among which the biggest challenge is to find the match between ERP functionality and business requirements or adjust system functionality accordingly.

The traditional method of resolving this problem is to form a team that will contain as many people as possible with knowledge of different components. But even high professional and experienced teams usually need to revert back to the training materials, help books, system help, use scenarios and other types of document that describe system business processes and configuration approaches. The problem here is that documentation analysis is usually time consuming activity because these materials contain some unstructured and not formalized information.

As an alternate solution we propose to use a special ontology-based knowledge management system that will be able to propose configuration according to customer's requirements. The approaches to modeling enterprise ontology have been already analyzed in some researches, but within our work we would like to use ontology not to model enterprise only, but also express the ERP functionality so that it allows to compare user requirements and system processes and to find missing configuration in the system. At this stage the proposed ontology based system has been already implemented for analysis and determining necessary ERP configuration but the biggest problem that was met within the ontology design is the determining correct approach for requirements formalization.

In our previous research the ontology for requirements formalization has been based on segregation of four parts in the requirements: trigger, actor, action, condition. Practical evaluation has shown that this approach doesn't allow analyzing dependency in the requirements.

Within this paper we would like to present an alternate approach that has been found to formalize the requirements: as the base of our requirements formalization method we propose to formulate requirements in the business process models, and map them to the ERP functionality using the Ontology for Linking Processes and IT infrastructure (OLPIT) [3] developed by J. Brocke, A. Braccini, C. Sonnenberg, and E. Ender.

The key benefit that it brings into the proposed knowledge management system is the linking between requirements and system functionality, and possibility to model dependencies both between system functionality and requirements. As a result it allows us to use the proposed system as an approach for automation of fig gap analysis process that, to the best authors' knowledge, is not possible with current OTS methodologies and tools.

Section 2 of this paper in the form of case study explains the main concepts of the initially proposed ontology. The purpose of that section is to show how we represented requirements on the first stage of the research and what challenges we met. Section 3 is devoted to the overview of different ontologies used for formalization of requirements or business processes. Section 4 contains the detailed description of the ontology for linking processes and IT infrastructure (OLPIT) that we consider to be the most useful approach of mapping IT functions and business processes. In Section 5 we present the new version of our configuration search ontology that includes ideas from OLPIT ontology. In Section 6 we conclude the results and determine strategic directions of future research.

2 Case Study of Ontology Application for Configuration Determination

The specific problems of ERP implementation projects seem to be very difficult and practically non resolvable. But if we analyze these challenges from developers' or integrators' view point, we can find simple and logical reasons for them.

First of all, current ERP systems provide a numerous number of functional scenarios and constantly enlarge the volume of functionality. For example, within Customer Relationship Management (CRM) system provided by SAP AG Corporation, the number of business scenarios is more than 300, and overview training about this product takes 4 – 6 weeks.

But despite the growing volume of ERP functionality, many organizations still can't find corresponding processes. To make ERP solutions more flexible, vendors provide different ways of systems configuration and modification. On the one hand, it gives a very useful opportunity to the customers to adjust system functionality to the user requirements. But on the other hand, it makes the process of implementation more complex because very often to enable some functionality (like orders search or campaign management) it is necessary to pass through several configuration scenarios, that are interconnected and data dependent.

But in spite of a variety of configuration methods that are provided within modern ERP, many companies still have to do system modifications. It is a third reason for many ERP specific challenges. Usually by modification we mean some changes to the system logic, user interface or database that are done though coding, database redesign or integrating new system self-developed modules on the same platform. Even nowadays modifications are still required for many companies. The survey [12] showed that about 65% organizations had to make some system modifications to meet end user requirements and about 50% of companies developed their own add-ons.

The problem of modifications is that ERP suppliers do not guarantee that upgrade packages won't destroy customer-developed functionality. The upgrade package can change business process logic, database scheme, programs signature or even whole user interface screen. Appliance of such upgrade packages is not mandatory but sometimes ERP vendors do not support system of very old versions and require system upgrades. In this case the only thing that can be done on customer side is a detailed regression testing that is very time consuming.

Another problem of modifications is that support services of big ERP vendors usually do not provide help if there are any problems with your custom developed programs but not with native functionality. Customer is not able to rely upon vendor's support and needs to have his own support team who can manage modifications and provide help if necessary.

From the authors' opinion, these challenges are the key factors of the ERP implementation projects failures. They make the process of ERP integration more and more difficult and put new questions and tasks before a project team. The project team needs not only to satisfy user requirements by developing the corresponding system features but also to answer a group of specific ERP implementation questions:

- How is it possible to use native 'Off The Shelf' (OTS) provided functionality as much as possible to build solution that will fully satisfy customer's requirements?
- If it is not possible to use OTS, what ways of configuration can be used instead of modifications?

These questions are not difficult from the first point of view but that is a serious problem in each particular practical cases. The main challenge here is to collect people with all required knowledge to solve the technical issue and investigate it from different prospective.

From the authors' prospective there is another solution to the problem of knowledge collection. We consider that it is necessary not only to build the team of experienced specialists but also provide them with knowledge management or decision support system that can help to retrieve required information more quickly than manual document analysis and to find most effective configuration approaches. From functionality prospective the proposed system should:

- contain structured and formal description of ERP functionality;
- contain structured and formal description of configuration methods and scenarios;
- be able to propose configuration according to customer's requirements.

So in general the proposed knowledge management system should represent the ERP system itself but in the formal and conceptual way and be able to do logical analysis comparing user requirements and ERP functionality.

To illustrate how an ontology-based knowledge management system can be used within implementation of information management system we have developed a prototype of such system for Customer Relationship Management (CRM) solution provided by SAP Corporation. Figure 1 depicts main concepts of the ontology that has been built.

The main concept of the proposed ontology is 'Capability'. By 'capability' we mean any feature or piece of system functionality that is available for the end user. For example within account and contact management scenario the SAP CRM system provides end users with capability to create, maintain and export business partners; within campaign management scenario users have ability to create, maintain and delete marketing campaign and trade promotions; within account identification scenario the system gives capabilities to find business partners, find installed bases and view last interactions. Any capability can be represented by two elements: the business object and the action. The 'business object' is a concept which is used to describe main

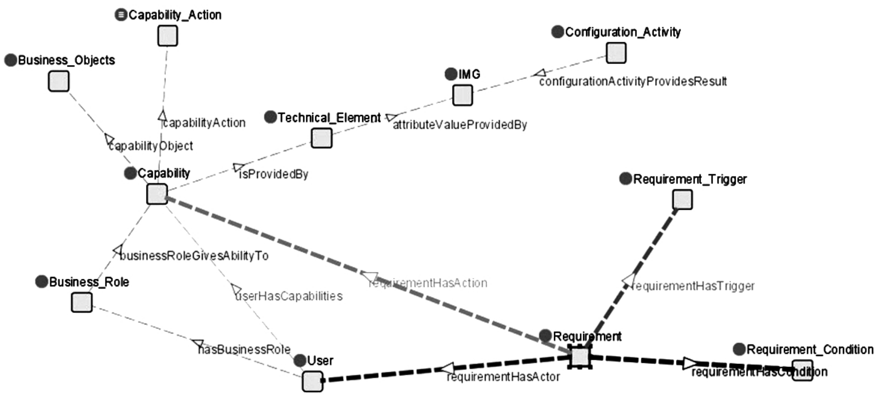

Fig. 1. Main concepts of the initial ontology for configuration determination

entities in the system which have business meaning for the end user. Separate sub ontology has been built to represent business objects that are available in the SAP CRM system and their relationships. In the examples of capabilities provided above, the business objects are ‘business partners’, ‘campaigns’, ‘trade promotions’, ‘installed bases’ and ‘interactions’. The actions that users can perform with the business objects are captured in the ontology as ‘capability actions’. Technically there are not so many actions that are provided by the information systems for the end users. We distinguish seven main actions: view, create, add, edit/modify, load, find and delete.

A ‘capability’ can exist only if it is provided by some technical object. By the ‘technical objects’ we mean all elements of the described systems that cannot be treated as business objects. Technical objects include user interface elements, transactions, database tables, function modules, programs, implementation guide objects and some other entries. Capabilities can be provided only by user interface elements or transactions.

Also a ‘capability’ can be provided by the system only if corresponding implementation guide objects have been configured. Implementation guide objects (IMG) allow developers to set additional system setting and change system logic through that. These objects are organized into the hierarchy that is called Implementation guide. The main problem of using this guide is to find correct objects and define the sequence of their changing. To make these tasks easier we developed the sub ontology of this Implementation Guide that repeats the hierarchical structure but within each class we defined the business objects classes that are impacted by the configuration object. We also introduce classification for the configuration objects and within ontology classes we assign each class a category as following: user interface configuration, attributes lists configuration, attributes dependency configuration, data exchange configuration, BAdI implementation. This classification is not pretend to be full but it can help to find particular configuration objects according to general requirements.

To capture the settings that should be done in the implementation guide object we introduced additional concept ‘configuration activity’ which contains a reference to the IMG object and specific instructions what should be done through it to enable some capability. The instances of ‘configuration activities’ are created dynamically when the reasoning mechanism of ontology analyzes the requirements or instances of business objects and finds the gaps between necessary functionality and existing configuration.

To capture requirements to the system functionality we also developed ontology of main requirements elements. This sub ontology is built on two ideas: the traditional classification of requirements (high level/detailed requirement; functional/non-functional requirements) and the approach that well formulated requirement should consist of four objects:

- Trigger – when the requirement should be satisfied/when action is allowed.
- Actor – who should perform the action.
- Action – what should be done.
- Condition – what additional restrictions exist.

Within the proposed ontology ‘actor’ of the requirement is an instance of class ‘user’ that is used to represent different users’ categories. It is not a particular person who is going to use the system but it is a group of people who are going to perform the same functions.

As ‘actions’ ontology propose to use any of the existing capabilities that have been already defined. At this stage we do not analyze a situation when there is a requirement for the capabilities that do not exist in the system as proposed ontology is designed only for system analysis and configuration search but not for the analysis of business requirements and their mapping to the system functionality.

Configuration search mechanism has been developed on the base of reasoning and inference techniques. The biggest part of configuration determination procedures is designed through logical rules written on the base of Jess engine. Designed rules can be classified into following categories:

- Rules that describe dependencies between the attributes of business objects and implementation guide objects that are used to configure available values of these attributes.
- Rules that describe the dependencies between the values of different business objects attributes.
- Rules that describe the dependencies between different implementation guide objects.
- Rules that describe the required configuration to extend the business object structure depending on the values of a particular business object attribute.
- Rules that describe transferring logic of capabilities between technical objects.
- Rules that describe the required configuration of user interface elements to provide a specific capability for the user.

The most important challenge that was met during building configuration search mechanism was related to the analysis and formalization of ‘trigger’ and ‘condition’ parts of the requirements. These elements can have a wide range of values that cannot be clearly interpreted by the ontology inference mechanism. Due to that a lot of relevant information is lost during configuration search.

Another problem found during the research was caused by the fact that sometimes one system capability may be dependent on the other capabilities if they are realized through specific system components. For example, within SAP CRM system it is not possible to create a service order in the interaction center if you have not identified a customer, but it is possible to create a service order if you are using a web client. The configuration settings may really differ in these cases depending on the sequence of actions that user needs to go through to meet the goal.

3 Review of Ontology-Based Approaches for the Requirements and Business Processes Specification

Trying to address the problems described in the section above, we made a decision to redesign the part of the ontology related to the requirements formalization. The analysis of existing researches in the area of ontological requirements representation shows that an ontology can be used for both, to describe requirements specification documents [6, 13] and formally represent requirements knowledge [13, 16]. In most cases, natural language is used to describe requirements, e.g. in the form of use cases. However, it is possible to use normative language or formal specification languages which are generally more precise and pave the way towards the formal system specification. Because the degree of expressiveness can be adapted to the actual needs, ontologies can cover semi-formal and structured as well as formal representation [16].

Advantages of using ontologies for requirements specification is definitely the fact that in contrast to traditional knowledge-based approaches, e.g. formal specification languages, ontologies seem to be well suited for an evolutionary approach to the specification of requirements and domain knowledge [16]. Moreover, ontologies can be used to support requirements management and traceability [13, 14]. Automated validation and consistency checking are considered as a potential benefit compared to semi-formal or informal approaches providing no logical formalism or model theory.

But unfortunately we have not found any ontologies that allow to model all aspects of the requirements pointing any common structure. Due to this reason, we considered alternative approach for requirement specification as the business process models. Such approach is widely used within the integration methodologies of information management systems. Within this approach all operations of the company are describes as business process models and are compared with the functional processes that are available in the information system.

Building an ontology that can be used to formalize structure of business processes has been considered in many researches. One of the first and most unified ontologies in these areas was designed by John Sowa [15]. His approach is based on the classification of processes and there elements. His taxonomy includes the following characteristics that can be used to determine type of the process: discrete or continuous, linear or branching, independent or ramified, immediate or delayed, sequential or concurrent, predictable or surprising, normal or equinormal, flat or hierarchical, timeless or time bound, forgetful or memory-bound

In general, the process ontology built by John Sowa allows classifying existing process by the categories but does not allow building a decomposition of the process to analyze its structure.

Another interesting research devoted to the ontological analysis of business process structures can be found in the materials of TOVE project that aimed at development of a set of integrated ontologies for modeling all kinds of enterprises (i.e. commercial and public ones) [8, 9]. TOVE Common Sense Model of Enterprise included three levels: reference model with typical business functions (finance, sales, distribution, and administration), generic model (with such concepts as time, causality, space, resources), and concept model (e.g. role, property, structure). The main approach that is used in this project for building business processes ontologies is based on the Process Specification Language (PSL). PSL was developed at the National Institute f Standards and Technology to axiomatise a set of intuitive semantic primitives that is adequate for describing the fundamental concepts of manufacturing processes. As elements of business process ontology PSL includes such concepts as activity, activity occurrence, time point, object, ordering, parallelism, decomposition, resource contention, states and conditions, complex ordering relationships, etc.

Another attempt to formalize the business process structure as an ontology can be found within researches related to the Enterprise Ontology [7]. The Enterprise Ontology was designed 1996 and still is the one of the most useful approaches for the description of enterprises. It was developed without being implemented in some computer language first, and was rendered formal and implemented later. It provides the primitives to describe all the important aspects of enterprises, and thus also processes. Process is not a central notion in the Enterprise Ontology, but it contains pretty much elements that can be used for describing the structures on the business processes.

The central notion for the ontology construction task at hand to be examined in the Enterprise Ontology is not ‘activity’ but ‘activity specification’. Note the interesting “nonreference” of ‘activity specification’ to ‘activity’. The ‘activity specification’ is not defined by formally referring to ‘activity’. The Enterprise Ontology mirrors some of the introductory reflections that were presented in this deliverable, e.g. that the ontology constructions require thinking about how the process instances in the BR domain can be distinguished from each other. The overall process can be represented through more than 40 concepts that include such elements as process specification, activity, activity specification, t-begin and t-end, pre-condition, effect, doer, sub-activity, authority, activity owner, event, capability, skill, intended purpose, resources, etc.

But the most complex and structured business process ontology has been designed within SUPER (Semantics Utilized for Process management within and between Enterprises) project [4]. Its purpose was to develop a set of ontology and tools for Semantic Business Process Management. Semantic Business Process Management [11] is a novel approach to the business process modeling and reengineering. Its main idea is to combine Semantic Web Services frameworks, an ontology infrastructure, and Business Process Management methodologies and tools, and to develop one consolidated technology that will lift the translation between the two spheres to a new level of automation.

One of the ontologies that has been designed within SUPER project perfectly describes the main concepts of the business process regardless of their notation. This ontology is called Upper Process Ontology (UPO). It is based on the DOLCE+DnS Plan Ontology (DDPO) that is founded on a theory of planning and on existing research on semantic descriptions of plans. DOLCE is a foundational ontology, i.e. it is a specification of domain independent concepts and relations based on formal principles from linguistics, philosophy and mathematics (e.g. concepts: Endurant, Perdurant, Quality, Abstract) and it is designed for reference purposes. DnS is an extension of DOLCE which provides an ontological theory of context, by adding concepts like Situation and Description. The main concepts of UPO are similar to the already mentions elements: plan, activity, task, actor, event, goal, condition, capability, etc.

The common feature of all business process ontologies described in this section is the fact that they describe the process from the enterprise point of view trying to combine the business aspects on the organization into the process. For our purpose it does not seem to be relevant data as the purpose of our ontology is to represent the functionality of information management system.

Taking into the account the main goal of our knowledge management system, we would like to review one more ontology that has been proposed in the [3] for linking processes and IT infrastructure and that seems most relevant for our problem

4 Overview of OLPIT Ontology

Figure 2 depicts the structure of the Ontology for Linking Processes and IT infrastructure (OLPIT) indicating its classes (the grey boxes) and their relationships (the arrows). Each box contains the name of the class and its attributes. Sub-classes inherit attributes from super-classes. Inherited attributes are indicated by the (…) notation.

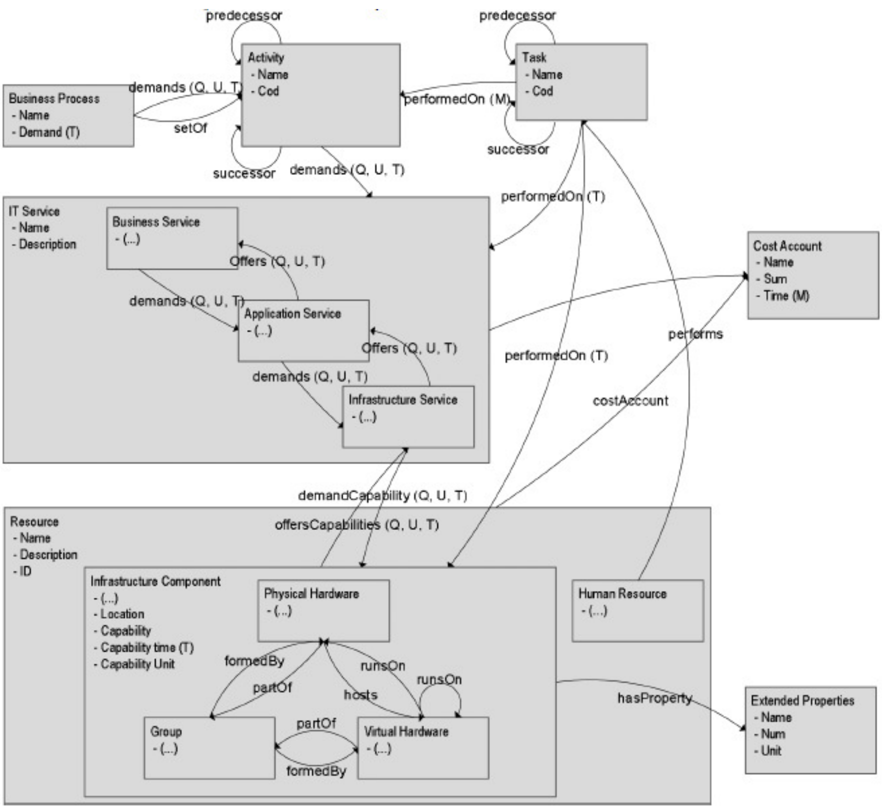

Fig. 2. The OLPIT Ontology

In the proposed ontology, the Business Process is the focal point. Business processes can be understood as value interfaces through which organizations deliver value to their (internal/external) customers. Following the implications of the thought of IS Alignment, the IT infrastructure delivers value to the Business Processes via IT Services. In order to be able to reason the structural relationships between IT components and business processes as well as to reason the course of value consumption (IT cost), the OLPIT ontology proposes classes and relationships of relevance in the application domain. Starting the description of the ontology from the bottom level, the IT Infrastructure is formed by IT components divided among hardware that can be Physical, Virtual or classified in Groups. A Group can be used to represent a set of hardware entities that are commonly interrelated (like for example a cluster of servers or the total ensemble of network components). In order to make the ontology schema general and not case dependant, IT Components can have extended properties associated to them (e. g. the amount of RAM, the amount of disk space, the amount of cache). IT Components, together with Human Resources, constitute the Resources that are necessary to deliver IT Services.

IT Services are divided into three categories: IT Infrastructure Service(s), IT Application Service(s) and IT Business Service(s). An IT Infrastructure Service delivers the capabilities of the IT Infrastructure Components to Application Services. Examples of such services could be a network service or a storage service. An IT Application Service is a service delivered by the functions of specific software. This class is not intended to include all software (e.g. operating systems) in an IT Infrastructure, but only those which are used to deliver services to the business side. Examples of IT Application Services could be e-commerce software, content management software, ERP software and so on.

Finally, an IT Business Service, is a service that delivers value to the customer side (via Activities and Business Processes). Under this perspective, an IT Business Service contributes to the execution of one or more activities in a process. An example of IT Business Services could be a credit card verification service. A Business Process is defined as a collection of Activities that takes inputs from other resources, manipulates them and produces outputs. Input and outputs may come from, or be directed to, other Business Process(es). An Activity may demand the execution of one (or more) IT Service(s) to deliver value or may require some Task(s) performed by Human Resource(s). Activities and tasks are linked in a chain and can have a predecessor and a successor. The capabilities of the IT Infrastructure and the demand of the business side are represented in the ontology by means of the quantity (Q), unit (U) and time (T) constructs associated to each demand/offer relationship. Finally, the proposed ontology models the cost information by means of the Cost Account class. A Cost Account represents a specific cost identified by its name (i.e.: depreciation), an amount (i.e.: €€ 1.500) and a time (i.e. year). Cost Accounts can be associated with IT Infrastructure Components, IT Services and Human Resources.

5 Adopting OLPIT for Configuration Determination

Despite the fact that the goal of OLPIT ontology is different from the ontology that has been built by us for configuration search, some similar elements can be found.

The first similar element that is used in both ontologies is ‘capability’. It is not formed as a class in the OLPIT ontology but it is mentioned by the means of the quantity (Q), unit (U) and time (T) constructs associated to each demand/offer relationship.

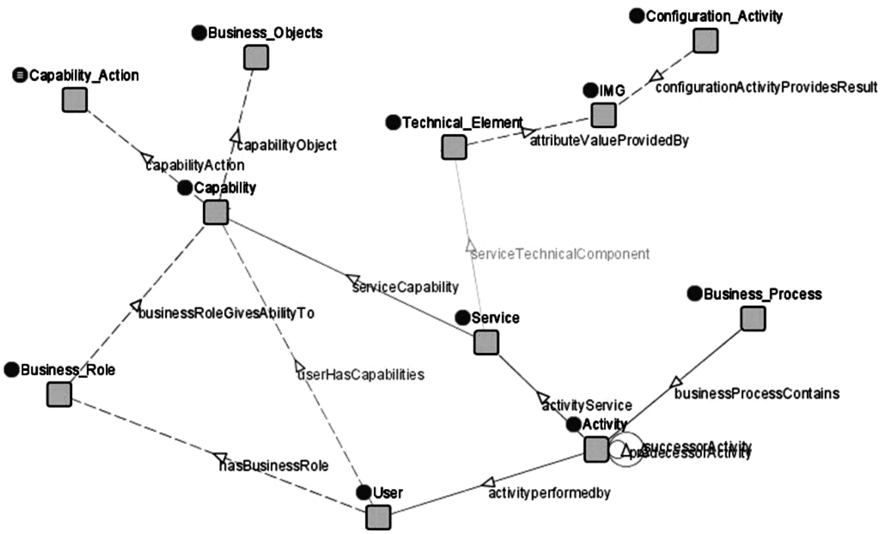

Fig. 3. The ontology of configuration determination based on the OLPIT concepts of business processes and services

Another similar concept that is used in both ontologies is Resource or Technical Element. In both ontologies by this concept authors mean some systems components that provides services or system capabilities to the users. In our ontology we are not interested in hardware aspects of the system structure but from the structure prospective we can say that resources and technical elements play the same role in the both ontologies.

Also it is interesting to compare the concept of the ‘service’ that is used in the OLPIT ontology with the concept of the ‘capability’ that is used with the proposed configuration ontology. From the first point of view, they are different. But from the conceptual point of view both concepts represent the functionality that can be offered to the users by the information system. The advantage of using the concept of ‘service’ is that we can separate technical components and system capabilities. It is more natural because very often one system capability can be enabled only by a set of technical components or one system capability can be provided by several components. In the case when there are direct links between component and capabilities the described situations become indistinguishable.

Also the biggest difference of the OLPIT ontology is the fact that for representing of business needs the authors propose to use the pretty easy notation of business process that is based on the assumption that business process consists of the activities which can be linked to the IT services. In our case of building an ontology for finding the required configuration according to the business needs we decided to reuse the

OLPIT approach and introduced the concepts of 'business process', 'activity' and the 'service' into the ontology.

'Business process' and 'activity' concepts have replaced the concept of 'requirement'. By this change the challenges of formalizing the dependencies between the system capabilities and unclearness of 'trigger' and 'condition' statements have been resolved.

6 Conclusions and Future Plans

As the conclusion of the work, we would like to mention that requirements engineering is still a difficult problem for implementing enterprise resource planning systems. Traditional approaches of requirements analysis are not perfect and cause projects failures as a result.

The analysis of the OLPIT ontology and some alternative methods to build the ontological representation of business processes suitable for requirements engineering has shown that OLPIT ontology proposes a good approach for formalization of the requirements specifying information management system functionality.

The combination of 'business process' and 'service' concepts gives a ability to model user needs pretty flexible and close to the common approach of business process modeling that is traditionally proposed by implementation methodologies of information management systems vendors.

In future we plan to extend the proposed ontology to make it possible to model not only system requirements but also business requirements and map business requirements to the system functionality.

The results of the project T3-29.0, carried out within the framework of the Basic Research Program of the Higher School of Economics in 20010, are presented in this work.

References

1. Boehm, B.: Model and metrics for software management and engineering, pp. 4–9. IEEE Computer Society Press, Los Alamitos (1984)
2. Boehm, B.: Industrial software metrics top 10 list. IEEE Soflw. 4(5), 84–85 (1984)
3. Brocke, J., Braccini, A., Sonnenberg, C., Ender, E.: A Process Oriented Assessment of the IT Infrastructure Value: A Proposal of an Ontology Based Approach. In: Abramowicz, W. (ed.) BIS 2009. LNBIP, vol. 21, pp. 193–204. Springer, Heidelberg (2009)
4. D.1.1. Business Process Ontology Framework. SUPER Deliverable (2007)
5. Daneva, M.: Lessons Learnt from Five Years of Experience in ERP Requirements Engineering. In: Proceedings of the 11th IEEE International Requirements Engineering Conference (2003)
6. Decker, B., Rech, J., Ras, E., Klein, B., Hoecht, C.: Selforganized Reuse of Software Engineering Knowledge supported by Semantic Wikis. In: Workshop on Semantic Web Enabled Software Engineering, SWESE (2005)
7. Dietz, J.: Enterprise Ontology. Springer, Heidelberg (2006)
8. Fox, M., Barbuceanu, M., Gruninger, M., Lin, J.: An Organisation Ontology for Enterprise Modeling. In: Simulating Organizations: Computational Models of Institutions and Groups. AAAI/MIT Press, Menlo Park (1998)

9. Fox, M.: The TOVE Project: A common-sense model of the enterprise. In: Belli, F., Radermacher, F.J. (eds.) IEA/AIE 1992. LNCS, vol. 604, pp. 25–34. Springer, Heidelberg (1992)
10. Gulledge, T.: ERP gap-fit analysis from a business process orientation. Int. J. Services and Standards 2(4), 339–348 (2006)
11. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In: IEEE International Conference on e- Business Engineering, pp. 18–20 (2005)
12. Kumar, B., Maheshwari, V.: An investigation of critical management issues in erp implementation: emperical evidence from canadian organizations. Technovation 23(10), 793–807 (2003)
13. Lin, J., Fox, M., Bilgic, T.: A Requirement Ontology for Engineering Design. Enterprise Integration Laboratory. University of Toronto (1996) (manuscript)
14. Mayank, V., Kositsyna, N., Austin, M.: Requirements Engineering and the Semantic Web, Part II. Representation, Management, and Validation of Requirements and System-Level Architectures. Technical Report. TR 2004-14, University of Maryland (2004)
15. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove (2000)
16. Wouters, B., Deridder, D., Paesschen, E.: The Use of Ontologies as a Backbone for Use Case Management. In: European Conference on Object-Oriented Programming, ECOOP 2000 (2000)