ACTA UNIVERSITATIS AGRICULTURAE ET SILVICULTURAE MENDELIANAE BRUNENSIS

Volume LX 20 Number 7, 2012

ENTERPRISE SERVICES AND OBJECT-ORIENTED METHODOLOGIES

M. Mišovič, I. Rábová

Received: June 22, 2012

Abstract

MIŠOVIČ, M., RÁBOVÁ, I.: *Enterprise services and object-oriented methodologies.* Acta univ. agric. et silvic. Mendel. Brun., 2012, LX, No. 7, pp. 179–188

In this paper we study the possibility of modifying the conventional, advanced object-oriented methodology for software development of enterprise information systems on an enterprise services platform. The presented solution is preceded with the draft of the formal description of the business process logic and its service solutions. In the design for the introduction of business services there is used the so-called native solution that is very close to the thinking of corporate management.

Adjustment of the classic, advanced object-oriented methodology is oriented on its cardinal development phases INSEPTION and ELABORATION. In these two phases important descriptions of relevant information modeling results are formed for the following CONSTRUCTION phase and TRANSITION phase. Editing, however, is methodological, and meta-documented with an example of a transparent graphical description of business services and a logical architecture of the target software.

In other words, it means that this contribution provides small software companies with valuable knowledge for the creation of their own development methodologies on a platform of enterprise services. The purpose of this paper is to contribute to the formation of implementing development methodologies, which absorb current knowledge disciplines of Software and Information Engineering.

object methodology, process logic, process threads, business services, service-oriented enterprise, business service specification, analysis and design of enterprise services platform, target software architecture

In the last two decades we have seen a significant shift in scientific areas related to the development of enterprise information systems. We mean especially the shifts in such disciplines as Information engineering, Software engineering, as well as in Theory of Enterprise business. The production of current enterprise information systems is directly dependent upon improvements the said disciplines have achieved. The Theory of Enterprise business already understands the enterprise business as a coherent whole of methods, techniques and tools that were used in the organizational enterprise construction, its information infrastructure, and of course the corporate information system. Information Engineering has closely followed the advances in information systems architectures. Software engineering focuses on advanced software development object methodologies.

Generally there is no doubt that the care of an enterprise about business processes, i.e. their optimization and flexible changes, brings the company a certain competitive advantage in the business environment of the dynamic predatory market. However, flexible implementation of these changes was in the application architecture of any enterprise information systems very demanding. In connection with the processes we often talk about the business process logic, the productions of which are complete process threads, very well researched by the company management. In connection with the business process logic objections have been gradually appearing that processes are not due to advances in Information and Software Engineering the most appropriate units of business functionality. In Theory of Enterprise business the idea has appeared to build a corporate view of the enterprise functionality by means of enterprise services. Then the functionality of each organizational unit of the company is based on its own services for the benefit of this unit and services for other organizational units. Quite naturally, it would be to understand an elementary separate business service as a process thread of the classical enterprise business process logic.

The understanding of the enterprise business process functionality based on the process logic has moved to a new platform, a platform of enterprise services. However, finding the business process logic transformation into a new service solution of its functionality has not been widely analyzed yet. The main role in this transformation has been entrusted to the company management and no to semantic relational systems that are to become productive business management assistance and the basis for the design of the business process logic transformation into its service solutions have not been looked for.

The business concept based on enterprise business services (SOE – Service Oriented Enterprise) has significantly changed the concept of enterprise information system architecture. The classical software application architecture of the enterprise information system has been replaced by SOA – Service Oriented Architecture, rapidly established in the discipline of Software engineering. SOA has enabled to implement unexpectedly flexible changes in business services. These changes are very important for the sustainable enterprise life as regards the business environment and the rapidly growing consumer market.

On the other hand, the concept of business services has combined with the Internet functionality, which with the aid of so-called Web Services, implemented its basic attribute, high distribution. In the course of time SOA has been of use even in robust software architecture for enterprise information systems.

Since SOA allows high modifications flexibility of computerized business services and their distribution, it comes in sight that the concept of services can be further manipulated, and enables their composing. Now not only the process threads given by the classic enterprise process business logic, but also their large sets, such as ERP, SCM, CRM and more are considered as a service.

SOA has a large paradigm, with many specific graphical notations. Additionally, SOA is not inconsistent with the object paradigm, it respects all three pillars of object–oriented thinking. Any service is seen as a class that generates instances for customers to use. SOA is also not contrary to the philosophy of a new approach to software that is known as Cloud computing. This philosophy is being built on software renting as a service and uses many different business models. The concept of service in this philosophy has not only greatly expanded to the entire ERP, CRM, SCM process sets but also to the whole complex information system and even moved from software to hardware. Then robust services such as ERP, data warehouse, information infrastructure, and others are rented.

Information engineering has gradually built the current software development of business information systems by means of very modern and advanced object–oriented methodologies (such as Rational Unified Process, Unified Process, Select Perspective and others). These methodologies are already using a single graphical notation UML – Unified Modeling Language for the results of information modeling description. The rapid development of SOA has caused that Information engineering does not accept some modern modeling methods of SOA paradigm projection. Instead, the paradigm of SOA has appeared as a separate section of the Information engineering, with its own methods, techniques and graphical tools for modeling of enterprise services life. Thus development methodologies constantly stay on the platform of business processes derived from use cases.

METHODS AND RESOURCES

Business process logic and its service solutions

Any manufacturing company has a control logic and business (application) logic. Both have a coherent logic to share enterprise life and are based on control processes and application processes. For our further considerations it is sufficient to use only the term process and specification of processes will be used later on.

A commercial and manufacturing company, in accordance with its manufacturing and sales focus on the customer, has a finite set *P* of business processes. Any process is understood by the company management as a comprehensive enterprise business activity unit with many of its sub-characteristics, corporate context and semantic associations to other business processes.

The business process context is established by the organizational unit that cares about the process and is responsible for its necessary quality. Semantic associations of the process to other processes provide important context of algorithmic associations, data, events, and other potential successive events and more. And especially significant semantic associations of enterprise processes are constantly examined by the company management and their knowledge is its knowledge property.

The concept of business process logic

Let us look at the concept of business process logic in terms of Knowledge engineering. It is not fantasy to think that just a set of mutual associations and business processes is the foundation of the business process logic, process logic *L*. This can be understood as a set of production rules, based on the identified associations between processes, producing coherent process threads of them.

The basis of any business process logic, however, is the association between two business processes with the possibility of generalization to a finite number of processes. Therefore we introduce the concept: elementary process thread.

Definition 1

Be *p*, *q* two different business processes. Be *A* a set of all associations between business processes identified by the company management. Let *R* be one of the subsets of *A*, thus $R \in \mathbb{P}(A)$. If for processes *p*, *q* the association *R* is valid, which we write *p R q*, then the ordered pair *(p, q)* is an elementary thread produced by the business process logic.

Although the elementary process thread is a significant result of the business process logic production, the company management knows and can manage process threads far higher than 2. To produce such process threads we suggest – for the business process logic – three operations **Create prefix**, **Create suffix** and **Create insert** that can freely expand process elementary threads, thus increasing the number of process threads dimension.

Be *(p,q)* a process thread. Let the associations R_1 be valid for processes *r, p* with production rules leading to the thread *(r, p)*, then the operation **Create prefix** is able to create a thread *(r, p, q)*.

Be *(p,q)* a process thread. Let the associations R_2 be valid for processes *q, r* with production rules leading to the thread *(q, r)*, then the operation **Create suffix** is able to create a thread *(p, q, r)*.

Be *(p, q)* a process thread. Let the associations R_3 **be valid for processes** *p, r* with production rules leading to the thread *(p, r)* and R_4 associations between the processes *r, q* with production rules leading to the thread *(r, q)*, then the operation **Create insert** is able to create a thread *(p, r, q)*.

These **o**perations **Create prefix**, **Create suffix** and **Create insert** are possible interpretations of specific production rules of the associations R_1 R_2 R_3 and $R_{4.}$ The **s**ufficiently introduced nature of the business process logic allows us to express its general definition. Denote the set of all business process threads by symbol *S* and all its subsets by notation $\mathbb{P}(S)$. The set $\mathbb{P}(S)$ is not generally disjoint. In other words, the process threads contain at least one common process. The preceding idea already allows us to formulate a general notation of the business process logic.

Definition 2

Be *P* the set of all business processes, *F* a set of all production rules and *A* a set of associations between business processes accepted by the company management. Then we call the business process logic trio *L = (P, F, A)* that is capable of forming a company process threads of the set *S* by using production rules and the identified associations between business processes.

Understanding the associations between business processes is far from completed business management knowledge. There is known that a research of interrelations between process threads and their orchestration inside the system based on them is not completely finished. On the other hand, the rugged and dynamic consumer market makes any company introduce new business processes, and there is always necessary to find their position in the already established process threads, or to create a new process thread for them. There is obvious that the system of threads is very much alive, and is flexibly modified. Given that our goal is a more advanced platform of enterprise activity, a business service platform, we will not deal with process threads problems consistently.

Although the information modeling of individual processes can use the UML language activity diagrams, process threads should be modeled by a more transparent and widely more capable technology. The appropriate technology is a proven workflow, i.e. network graphs on the platform of features and possibilities of Petri nets.

Service Solutions of business process logic

In (Mišovič, 2007) is the emergence of corporate services firstly characterized from the position of informatics and then from the view of theory in terms of business. Many computer programmers and IT managers understand the emergence of service-oriented architecture of an enterprise information system as a necessary response to solutions disability of flexible process changes in old software application architecture. In many articles, authors of which are primarily computer programmers, the service orientation of the enterprise information system functionality is attributed only to enterprise software computerizing the enterprise activities. There is believed therefore that it is naturally some software feature and not the company itself. The company then has to adapt itself to this architecture.

But the problem is more complicated. The already existing enterprises / systems based on the consumption of services given by other enterprises / systems are being forgotten.

Now there is a question if the enterprise functionality could be built on mutual services between the departments (or divisions / sections), including the understanding that the department can provide certain services to itself. It will be a kind of practice (with given rules) of providing business services in the territory of any enterprise. It is possible, though vaguely, to understand any enterprise service as a unit that can be composed of discrete business functionality, or composed from a certain number of interrelated business processes.

Introducing the concept of ESA (Enterprise Service Architecture) is confirming the previous idea that has a revolutionary impact on the architecture of ERP set, see (Woods, 2006). Enterprise services are generally introduced, however vaguely and naively by means of next ideas:

1. The entire functionality of the company is divided into services that are potentially reusable and are an integral part of enterprise functionality, i.e. for example transaction/process, or if you like another enterprise activity (control, application-business).
2. Enterprise services are independent. Business services can be grouped into higher units that can be considered as new business services.
3. Enterprise services form a system of corporate services providing the framework for mutual cooperation (orchestration).
4. Descriptions of services with the framework for cooperation – orchestration are freely accessible not only for the management of enterprise services administration.

For the purpose of non-vague introduction of enterprise services can be used the possibility to transform the business process logic in its service solutions. Of course, there are several bases for this transformation, for example:

1. Native transform basis that is based on process threads produced by the business process logic.
2. The specific semantic systems for creating enterprise services, aside the enterprise process logic.

Native transformation basis is the closest the company management. It is characterized by the fact that it utilizes the results of process logic that produces threads (control, application–business), much more complex entities than individual processes. The basis of the mentioned transformation is given by the following definition.

Definition 3

Be $L = (P, F, A)$ any business process logic. Each thread v of the processes produced by process logic L will be considered as a *fundamental enterprise service s*.

The specific semantic systems based on different types of associations between business processes beyond the concept of native business process logic, give the possibility of very specific enterprise services. For example, services based on application of artificial intelligence methods for business management, enterprise production, etc.

A workflow model for enterprise services

The initial focus is the greatest use of workflow technology especially in companies where it serves as an information base in which sources, documents, and their movements (distribution, etc.) as well as assigning users to specific tasks can be observed. We can manage all the company processes with the aid of their functions, we can monitor processes, manage the document flow and the gradual process of their completion and approval, as well as supporting software applications management. This workflow technology, in this sense, is inseparably connected with the business process logic.

On the other hand, workflow technology is successfully usable also for the graphic description of business services, which essentially integrate certain processes. Workflow forms a transparency for business and manages existing services and also helps to simplify them, improves the efficiency and shortens the entire service in number of its processes. We can generally say that automation of business services raises new requirements. Let us mention for example, increased demands on precision and clarity of specification processes in the service. The user is then able to interpret the definition of business services, interact with workflow participants and, if necessary, to start another service or perform the defined action.

From a technical perspective, workflow technology is very interesting and combines principles, methodologies and technologies of various areas of computer science and management – the concept of client / server, electronic mail, database processing, task management, documents, knowledge, modeling and monitoring business processes in service, etc. As such, it is a suitable candidate for solving of a graphical description of business services.

Petri nets (Petri Nets – PN), which are essential characteristics of workflow technology, is a proven tool for modeling systems in parallel with the ongoing processes. Similar systems also include business services. The advantage is the possibility of a transparent graphical representation of the modeled system services. They allow an accurate description of the structure and behavior of the modeled system. PN analysis enables to determine significant features of the model (and thus the modeled system) that can be used for its optimization.

Time PN expansion enables to work, among other things, with minimum / maximum duration of the transition or the minimum residence time of markers in each location. Business services last non–zero time, so it is advisable to use a time extension. Timing space will serve us in the case where between the two successive processes in the business service must be a minimum delay.

Potential benefits of Petri nets for modeling business services

PN have their own network graph from which one can see not only following sub-processes in the business service, but also the conditions of their implementation as well as their impacts. Graphs of large enterprise services can be a hierarchy (divided into several hierarchical sub–graphs) and thereby increase their visibility. Properties of the model, found by means of the analysis, can provide valuable information about the modeled system workflow. Properties of transitions may reveal an illogical set of business rules which cannot meet the conditions for carrying out certain operations in the business service process.

For our case, the use of modeling workflow for management services in the corporate system, Petri nets are suitable for the following reasons:

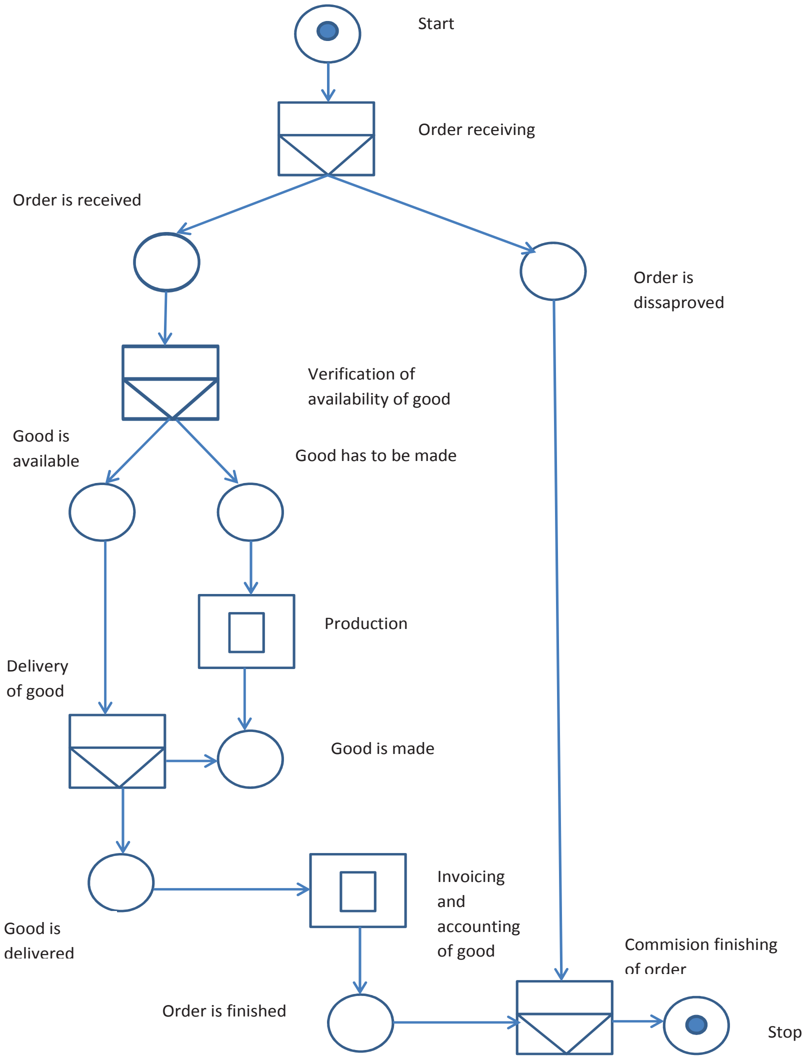
1. Formal semantics of the graphical notation – control logic can be described graphically and formally yet. Formalized is even the addition of several Petri nets (color, time, hierarchy) that can be used for the specification of individual services within the workflow for the phase of the model design and analysis, especially business services. Due to the high formalism the model is unique and can be used for the definition of contracts, is independent of the tools, is useful for assessing the performance of services and offers the use of analytical techniques.
2. Petri nets are based primarily on states rather than events, as in other modeling techniques such as UML. The sequence of individual activities interspersed with changing states of objects can be included in business services and can precisely define the service.
3. You can add other attributes of the service, such as the parameters of the elements in the Petri net. Classical Petri nets are not well suited for modeling workflows, they are semantically poor. It is therefore necessary to use high level Petri nets, which allow even parameterization.

The use of workflow technology for the graphic description of a business service is illustrated in Fig. 1.

RESULTS AND DISCUSSION

Enterprise services as a new platform for advanced object development methodologies

The reality of contemporary software development of enterprise information systems is a partial implementation of advanced object–interaction methodologies, currently SOA paradigm. It is therefore reasonable to submit a general proposal to



1: *Using of workflow technics for a graphical description of the service „Order administration“*

change the object-oriented methodology platform of business process to the platform for enterprise services. Thus influenced object methodology fully respects the concept of the kind SOE, as this concept is introduced in (Woods, 2006). Of course, this can greatly affect the development of the concept of architecture development of the target software. All this can also lead to new graphic descriptions of information modeling results, or extend those that are already part of UML.

The first use of the SOA paradigm results was focused on the transformation of the now obsolete software application architecture, enterprise IS, see (Woods, 2006). Only the development of new IS has led to the consistent application of the basic features of SOA and its paradigm, see (Erl, 2005). This has led to strict compliance with SOA as an architectural model of the new ERP software with high flexibility with distribution. Business services are now considered as basic control or application components in the architecture of software, whereby it has become to some extent the approach to classical object-oriented methodologies.

This chapter presents a proposal which new workflows should be added to the classical object-oriented methodology, or how its current working practices transform, especially at the relevant stages INCEPTION and ELABORATION to the software development platform with dominating enterprise services. We will not deal with stages CONSTRUCTION and TRANSITION.

Conventional advanced object methodologies use approximately about twenty working procedures. We will only deal with some relevant of them (Requirements, Analysis and Design), which are by their content directly connected with the dominance of the classic use cases based on business processes at the stages INSEPTION and ELABORATION.

New workflows, which we are going to propose, are described using only simple meta-patterns which include the specification of input and output artifacts. However, this chapter can also serve as a guide for software companies how to suggest a structural design of implementation development methodology on the platform with enterprise services.

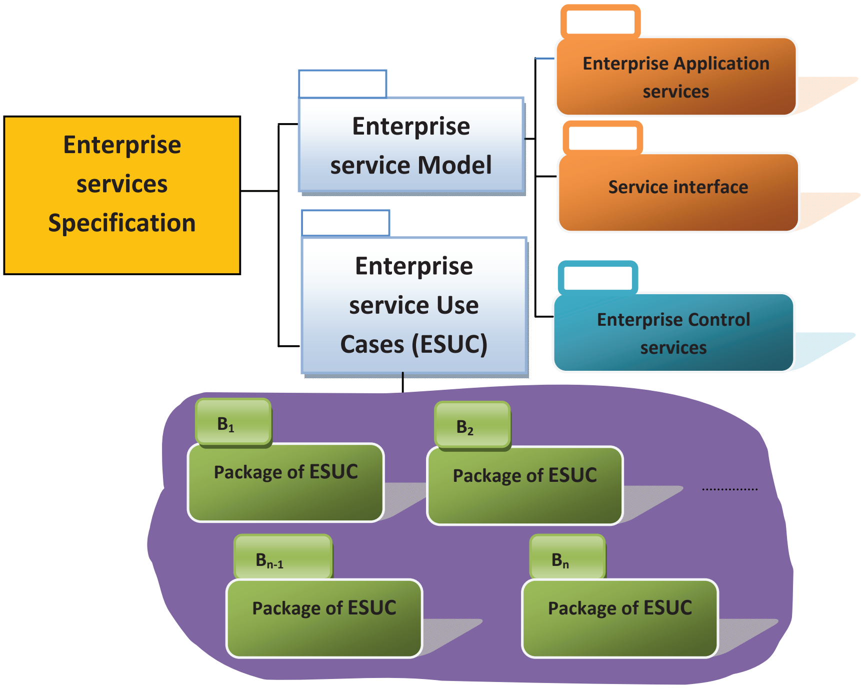
The phase INSEPTION

The basis of this phase in object methodologies is the dominance of business processes that are in the *workflow requirements* mapped to *Use cases*. Use cases are then always between the input artifacts of working procedures in the next stage of ELABORATION.

We introduce a business service platform through the *workflow enterprise services*, which will replace the workflow requirements. Workflow enterprise service has four major working steps:

1. Analysis of application (business) enterprise services.
2. Analysis of control enterprise services.
3. Mapping of enterprise services to their Use cases of services.
4. Analysis of the interface between two layers of enterprise services.

The aim of the first working step is to find application enterprise services for each semantically broader and more comprehensive field of business activity and make their brief verbal description.



2: *The meta-model of an Enterprise services specification*

These are the services forming a part of the business logic and they do not comprise any control elements. Application (business) service description specifies what part of the business logic each service encapsulates. The first step is manageable only with the direct participation of the company management.

The second working step is looking for control services for each semantically broader and more comprehensive field of business activity. These manage the use of the rich consistency of application services in this field. The advanced technique of workflow can be successfully used for the modeling of the interrelationship of the control and application services. This step brings the so-called *Enterprise service model*, a very important input artifact for the stage ELABORATION for the mentioned activity. This model allows flexibility and agility needed for changes of application and control services.

In the third working step there is carried out *mapping* of enterprise services to their business Use cases (*Use cases of enterprise services* – ESUC), *modeling* Use cases of enterprise services, including *packaging*, and *description* in more details. The graphical modeling of the main scenario for each Use case of enterprise services uses the advanced workflow technology. By this mentioned activity the specification of enterprise services is prepared and there are ready all input artifacts to the stage ELABORATION, see Fig. 3.

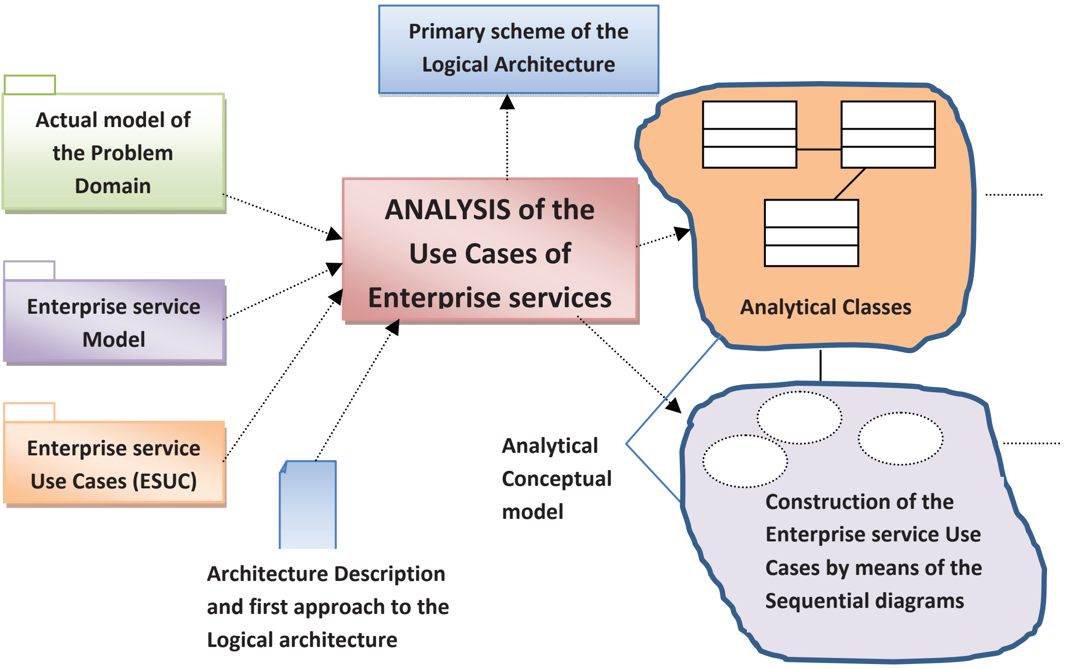
In the fourth working step there is carried out the analysis of some relevant connection interface between the business services and control services. Naturally, there is provided a possibility for flexible interface changes and then for rebuilding the relevant context between control and business services.

The phase ELABORATION

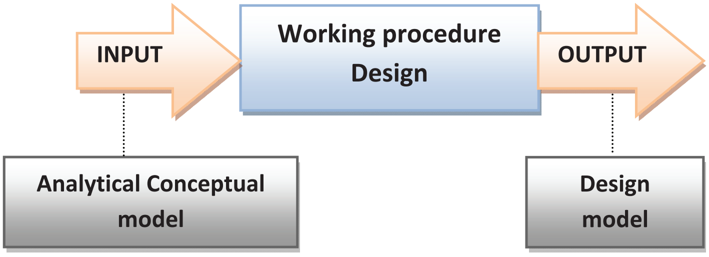
The phase Elaboration in the framework of the classical object advanced methodologies, with the dominance of Use cases, is devoted to the modeling analysis of the relevant diagrams of implementation classes, state diagrams and sequence diagrams. Orientation is observed on the actual implementation of individual Use cases through the use of operations and attributes listed in the implementation of analytical classes. The analytical model of exercise classes, state diagrams and sequence diagrams are eventually transferred to the design form. However, in addition, at this stage logical and design target software architectures are modeled and they have to accept also the basic properties of the given source problem domain. Advanced workflow technics are used in Analysis and Design of both of these architectures.

To implement an enterprise services platform we have to modify both of the input artifacts for Analysis and Design, see Fig. 3. We consider the specification of Use cases of enterprise services for the input to the Analysis, while the output will be an analytic conceptual model - the relevant object class diagrams, state, sequence diagrams and logical architecture of the target software.

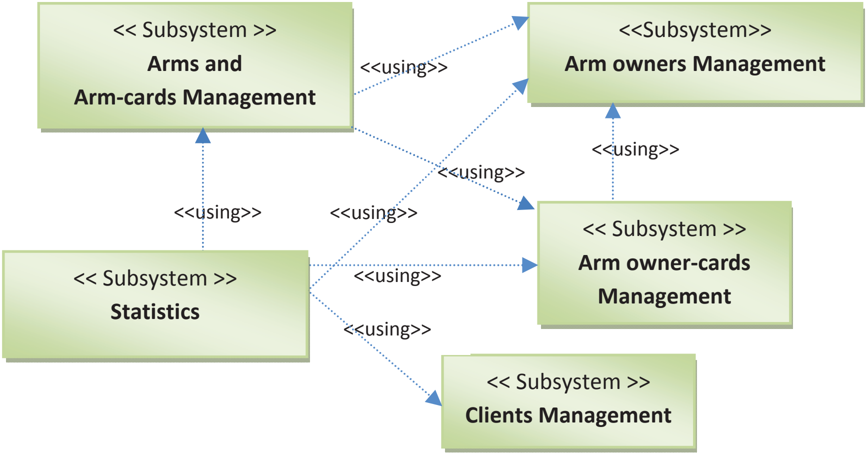
Workflow Design must be considered for admission of the analytical conceptual model, the output will be the Design model – the design form of object diagrams and relevant design software architecture of the software target. An application of


Primary scheme of the Logical Architecture
Actual model of the Problem Domain
ANALYSIS of the Use Cases of Enterprise services
Enterprise service Model
Enterprise service Use Cases (ESUC)
Analytical Classes
Analytical Conceptual model
Construction of the Enterprise service Use Cases by means of the Sequential diagrams
Architecture Description and first approach to the Logical architecture

3: *Use cases analysis, meta-model of business services*



4: *The workflow meta-model of the artifact Design*



5: *A primary access to the Logical architecture*

the business services platform and changes in both input artifacts Analysis and Design are illustrated in Fig. 3 and Fig. 4.

The semantically complete activities, such as a recognized set of process ERP, SCM, CRM and others can be understood as a problem domain structure model for the Actual model of the Problem Domain. The Meta-model in Fig. 3 is functional for each individual Use case of a business service, because it evokes the implementation of analytical finding classes, their attributes and operations. In addition, the Design evokes the realization of a single business service Use case through a Sequence diagram.

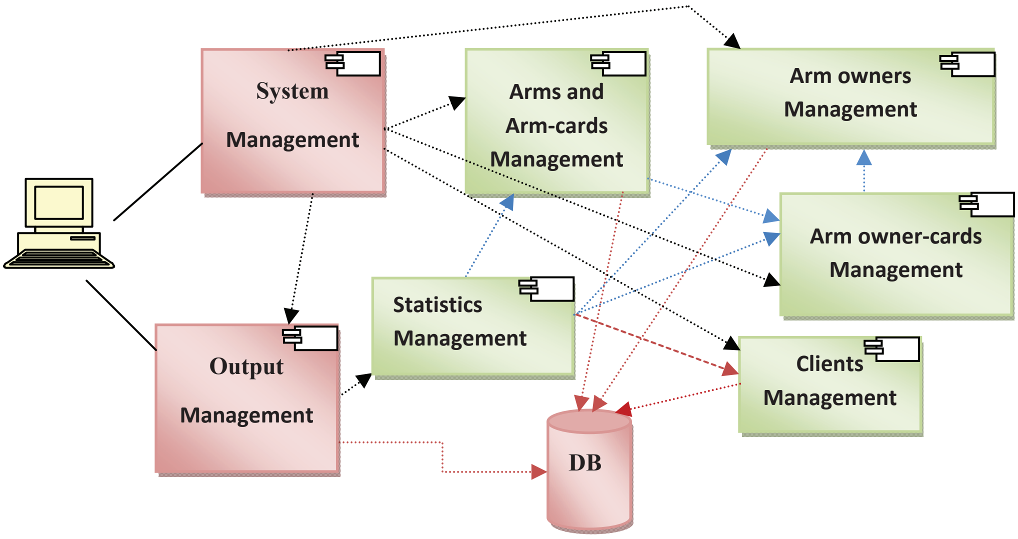
The stage ELABORATION also deals with the Design and Logical architecture of the target software. This is based on the initial approach to the architecture of the start-up. Of course, the architecture must be built on enterprise services. This is because the Logical architecture is based on relevant characteristics of the problem domain. On the other hand, its functionality is independent of the implementation of information technologies.

Although the Design architecture is based on the Logical architecture, primarily is under the influence of the chosen model of the information technology implementation. Design architecture is then easily transferable to the Physical architecture and then to the source code form. The initial access to the Logical architecture of the target software architecture usually presents the derivation of each sub-set of enterprise services, such as bound on relation <<using>>.

Illustratively, this idea is applied in the initial approach to the Logical architecture of the target software of the domain “Management of private weapons on the territory of the CR”, see Fig. 5.

The Logical architecture is not directly usable for the construction of the Physical architecture (constructed from Design architecture), but it illustrates only relevant characteristic of the problem domain. It is therefore necessary to supplement the initial approach to illustrate some of the relative attributes of the target software domain (context of the problem domain and its software domain). So we rename the parent subsystems to the robust components and add a component for internal control in the target software (communication with clients, starting robust components). We can add also other components to enable efficient handling of parent robust components (summary management reports and system control) and illustrate the data-driven internal control. Due to the broader transparency we can add to the initial access a computer icon for contact with the target software.

The scheme based on the previous ideas is already the first *Logical architecture diagram* for our primary problem domain “Management private weapons on the territory of the CR”. Subsystems are considered as robust components of the target software and it is respected that the architecture will use the data-driven technique.



6: *The first Logical architecture diagram*

In the workflow Design proposal we can map each robust component from the first Logical architecture diagram to collection of schemes. Components of each collection are already expressed in enterprise services and their interfaces respect internal and external collaboration. Naturally, we can thus get the first *Design architecture diagram* for the target software.

SUMMARY AND RECOMMENDATIONS

In this paper we present a modification of a classical, advanced object-oriented methodology in a way that it uses the platform for enterprise services for software development of an enterprise information system. We proposed a theoretical approach to the concept of business process logic, completed by formal notations based on the set theory. In addition, we designed the transformation of business logic to its service solution. For further considerations, we chose the native basis for the transformation resolution of any process logic to the business logic on enterprise services and we used corporate management knowledge.

As an extension we have pointed out the ability to create more complex business services than native services. We have not dealt with these algebraic capabilities in details. We have not touched systems of relationships among business services that enable their composition.

Our further considerations were focused on the transfer process of any object methodology to an object-platform on enterprise services. The transfer was examined only for the two cardinal phases INSEPTION and ELABORATION of any object methodology. Using a suitable meta-model we have shown the creation of Use cases of business services and their modeling in the phase INSEPTION. We have evaluated the using of transparent workflow techniques based on Petri nets for the description and strong refining of Use cases of enterprise services. By means of another meta-model and with help of object classes and sequence diagrams we have appointed the issue of Use cases of enterprise services implementation at the stage ELABORATION.

Our great attention was paid to the development of the Logical architecture for the target software and its using for the Design architecture. Our considerations are documented in a transparent example of the problem domain “Management of private weapons on the territory of the CR”.

Of course, issues of object-oriented application development methodologies on enterprise service platforms have to be solved in several directions so that the material becomes a good methodology for software developers themselves. There is no need to stress the philosophical nature relevance and severity.

Based on this philosophical idea, relevant theoretical considerations, the related methods, techniques and tools can arise. This philosophy must be flexible to enable the acceptance of the best practices, techniques and tools from both the SOA paradigm, the current object conventional methodologies.

REFERENCES

JACOBSON, Ibooch, G., RUMBAUGH, J., 1998: *The Unified Software Development Process*. Amsterdam: Addison Wesley Longman, Inc. ISBN 0-201-57169-2.

RÁBOVÁ, I., JEDLIČKA, P., 2011: *Using of the UML and Petri nets for optimization of Enterprise workflows*. SR, Nitra: Constantine the Philosopher University, Modeling methods and Data Analysis in IS. ISBN 978-80-555-0034-9.

KRUCHTEN, P., 2003: *The Rational Unified Process An Introduction*. USA: Pearson Education. ISBN 0-321-19770-4.

ERL, T., 2005: SERVICE ORIENTED TECHNOLOGY: CONCEPT, TECHNOLOGY AND DESIGN. USA: Prentice Hall. ISBN 0-13-185858-0.

WOODS, D., MATTERN, T., 2006: *Enterprise SOA. Designing IT for Business Innovation*. 1st edition. USA: O'REILLY®. ISBN 0-596-10238-0.

ERL, T., 2006: *Service-oriented Architecture. A Field Guide to Interacting XML and Web Services*. USA: Prentice Hall. ISBN 0-13-142898-5.

MIŠOVIČ, M., 2007: *Application Architectures of Enterprise Information Systems versus Service Oriented Architecture*. Acta univ. agric. et silvic. Mendel. Brun., LV, 6: 233–242. ISSN 1211-8516.

EELES, P., CRIPPS, P., 2010: *The process of software architecting*. USA: Pearson Education. ISBN 0-32-135748-5.

BELL, M., 2008: *Service-oriented Modeling (SOA) Service Analysis, Design, and Architecture*. USA: John Wiley & Sons. ISBN 978.0-470-14111-3.

Address

prof. RNDr. Milan Mišovič, CSc., doc. Ing. Ivana Rábová, CSc., Ústav informatiky, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, e-mail: misovic@mendelu.cz, rabova@mendelu.cz