

Advanced Approach to Future Service Development

Tetiana Kot¹, Larisa Globa¹, and Alexander Schill²

¹ National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine

² Technische Universität Dresden, Fakultät Informatik, Deutschland

Abstract. Modern companies, including telecommunication ones and mobile operators, working in the global environment, need to guarantee technological effectiveness and innovation, renewing their technologies and services. Communication technologies and variety of services are to be improved and developed extremely fast. This results in the need for constant adaptation and reconfiguration of complex software systems, used in global environment to provide and monitor deployed services. For instance, OSS/BSS is used in the domain of telecommunication companies. Reconfiguration of complex software systems covers application development for service provisioning, their dynamic monitoring and reconfiguration. Currently, system adaptation and service design strategies are poorly formalized and validated. In current state-of-the-art approaches, several iterations involving analysts and system architects are necessary, resulting in time and money consuming service development. The approach proposed in this paper fills this gap. It employs a well-defined workflow and analysis model for developing and adapting complex software systems. The applicability of this novel approach is confirmed by an implemented software tool. The proposed approach and tool provide automation of service development, focusing on planning and design stages, considering both functional and non-functional requirements and realizing computational independent workflow transformation into its execution model.

Keywords: service development, OSS/BSS, non-functional requirements, workflow design and transformation, Dia editor.

1 Introduction

Today, companies, working in global environment, constantly renovate and improve their technologies, providing services by web-oriented applications [1]. This is realized via service development and re-engineering. Companies tend to minimize time of service provision, constantly improving communication technologies and applications. Thus, means and tools for fast workflow design and reengineering during system runtime and for providing services within a minimal time are in high demand.

The lifecycle of service development is an iterative process, consisting of the following stages:

1. workflow design: development and description of computation independent and computational workflows, including service planning, applications prototyping;

2. workflow analysis and simulation, considering several system parameters;
3. workflow enactment, providing deployment models, considering both hardware and software service deployment alternatives;
4. service/workflows monitoring, leading to services re-configuration.

Currently, the service planning stage is performed manually. This does not result in optimal solutions because a lot of factors have to be considered when planning differentiated services [2]. Furthermore, in the current state-of-the-art computational independent workflow design, being performed using existing notations and tools, on one hand all required parameters, necessary on the service planning stage, such as numerical values of execution time and resources, and also document and information flows, supporting service provision, are not considered and on the other hand poor connection to system functionality, realizing these workflows are provide.

When service design and deployment, an important aspect concerns NFR¹ to service provision. It is extremely important to meet NFR on design stage, otherwise provided service may be useless in practice.

Currently, NFR are not considered with perspective of provided services list as legacy methods can design service according to NFR, but cannot model an influence of concurrent services on particular NFR because of collaboration between services. There are also no tools allowing flexible balancing between services.

Transformation processes between workflow modeling stages is still an open question, though there are transformation standards and languages [3], but this issue is not fully automated and has to be investigated to make service design correct.

This paper describes an advanced approach to future service development, including novel computational independent workflow design method, NFR balancing method and modified service workflows transformation method.

The paper is structured as follows: Section 2 contains state of the art analysis of workflow design notations and tools. State of the art analysis of methods and approaches to considering NFR. NFR analysis methods and service implementation technologies are described. BPMN to BPEL transformation strategies are also considered.

Section 3 introduces advanced approach to service development. It includes workflow design method, focusing on two core aspects: a computational independent workflow model and workflow analysis method, providing service provisioning time minimization; NFR balancing method, focusing on functional and non-functional requirements collaboration; modified service workflows transformation method.

Section 4 presents a prototypical realization of the advanced approach, including design and transformation tools and highlights evaluation results of the developed tools. The evaluation has been applied using a real-world scenario within a telecommunication company.

Section 5 concludes the work with a summary and outlook on future work.

¹ Non-functional requirements.

2 State of the Art and Background

2.1 Workflow Design Notations and Tools

Service planning is defined by finding a good way to create service provision at minimal time, having specific resource values as a limitation. In the following a summary of the state of the art of the central areas of this overall field will be discussed. A state of the art analysis of workflow design notations and tools, workflow analysis methods and systems, parameters, which should be taken into account at the planning stage and NRF design methods are presented.

Computational Independent Workflow Design

Computational independent workflows are designed using graphical standards, allowing their formalization in a diagrammatic way. Analysis has shown that in practice computational independent workflows are usually designed using graphical notations such as BPMN 2.0², UML AD³, USDL⁴ and tools such as CA ERwin Process Modeler⁵ and Enterprise Architect⁶.

The main criteria, used when analyzing current state-of-the-art notations and tools for computational independent workflow design, are possibility to consider and analyze several parameters, necessary on the service planning stage, such as service provision time and resources, and on the other hand provide connection to system functionality, which should realize these workflows.

USDL doesn't meet all the requirements of workflows analysis and design, its usage is difficult due to its complexity, in spite of its comprehensiveness.

The main argument against using regular BPMN is that management of resources can be expressed only via lanes (actors, roles, etc.) or performers of user or manual tasks [4], when execution time parameters are not considered. All further existing workflow modeling notations have this core criticism in common.

Nevertheless, BPMN, providing computational independent to computing workflows transformation (BPEL⁷ diagrams), was applied as a basic notation for computational independent workflow design, being extended with added missing concepts [5].

Workflow Analysis

The overview of the workflow analysis methods and tools has shown, that there exist two types of analysis, both considering computational workflow [6]:

1. Design time analysis (simulation and verification);
2. Runtime analysis (i.e., process mining, based on the execution logs).

² <http://www.omg.org/spec/BPMN/2.0/>

³ UML Activity Diagram.

⁴ <http://www.internet-of-services.com/index.php?id=264>

⁵ http://erwin.com/products/detail/ca_erwin_process_modeler/

⁶ <http://www.sparxsystems.com/products/index.html>

⁷ Business Process Execution Language.

Software tools such as Cactus, ASKALON, GLUE, etc. [7] are used for these analysis fields. Mentioned and analyzed current possibilities for this task stage are very limited. Shortcomings of workflow analysis methods and tools are clearly described in [8]. The central criticism is that the requirements analysis stage is applied mainly in a manual manner.

2.2 Methods and Approaches to Considering NFR

Early-phase requirements engineering should address organizational and NFR, while later-phase engineering focuses on completeness, consistency and automated verification of requirements. Not proper dealing with NFR leads to considerable delays in the project and consequently to significant increase of the final costs [9].

NFR are considered on design stage and there are several approaches to model NFR within the scope of developed service. NFR framework [10] is a methodology that guides system to accommodate change with replaceable components. Such NFR as security, accuracy, performance and cost are used to drive the overall design process and choosing design alternatives.

KAOS [11] is another methodology, considering NFR. It allows requirements engineering, enabling analysts to build requirements models and derive requirements documents from KAOS models.

There are some other approached and methods to NFR modeling [12, 13], but all of them don't consider collaboration between FR⁸ and NFR. The legacy software tools, such as NFR-Assistant CASE⁹, ARIS¹⁰, don't provide functionality to model NFR and consider their influence on system functionality.

2.3 BPMN to BPEL Transformation Strategies

There are various BPMN to BPEL transformation strategies, such as Element-Preservation, Element-Minimization, Structure-Maximization and Structure-Identification [14], Event-Condition-Action-Rules [15] and others. A basic idea of mapping which is used in most algorithms of translating BPMN or any other WF-net into BPEL code uses a mix of mentioned strategies.

Two categories of tools are applied in practice:

1. BPMN graph is serialized to an XML document. Then the last one is automatically translated into an abstract BPEL document. The abstract BPEL is enriched with the pieces of information, necessary to make it executable.
2. BPMN graph is translated directly into executable BPEL code. This is only possible when input/output files of the future Web Service (WSDL files) are created in advance.

⁸ Functional requirements.

⁹ Quan Tran "NFR-Assistant: tool support for achieving quality", Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium.

¹⁰ http://www.softwareag.com/corporate/products/aris_platform/default.asp

There are three main problems of BPMN-BPEL translation:

- fundamental mismatch between these two languages: BPMN is graph-structured notation while BPEL is block-structured language;
- readability of the resulting BPEL code is very low;
- translation of extended notations with additional workflow parameters is not possible.

Hence, development of the method providing automated translation of extended BPMN elements is an important aspect in service development process.

3 Advanced Approach to Service Development

Proposed advanced approach to future service development includes:

1. novel computational independent workflow design method, focusing on the workflow model and its analysis, allowing to automate service planning stages;
2. NFR balancing method, focusing on collaboration between functional and non-functional requirements, allowing to automate service planning stages;
3. modified service workflows transformation method, providing abstract to execution workflows transformation, considering workflows required parameters.

Each of the proposed method is briefly described below.

3.1 Workflow Design Method

Workflow design includes workflow modeling and simulation. The novel method of workflow design presented in this section is focused on computational independent workflow and consists of the following stages:

1. extended modeling of computational independent workflow;
2. computational independent workflow analysis, including:
 - 2.1 forming workflow graph and verifying its connectivity;
 - 2.2 workflow execution time minimization;
 - 2.3 transformation of workflow to realization diagram.

The suggested method modifies MDA¹¹ on the business logic level [6].

Workflow Model

The workflow model is one of the core aspects of the proposed method, allowing its formal description and thus its in-depth analysis and transformation to more fine-grained representations. In the following, the workflow formalization variant used within our approach is presented.

The mathematical formalization of a workflow can be done by using:

$$BP = (E, I, P) \tag{1}$$

¹¹ Model Driven Architecture.

where E is the set of workflow identification objects; I is the set of workflow informational objects and P is the set of workflow parameters of service provisioning.

The identification objects $\{E_{id, id=1,4}\}$ include: $E_1 = \text{name}$, $E_2 = \text{description}$, $E_3 = \text{executor}$, $E_4 = O$ – set of works.

The set of workflow informational objects includes income and outcome document and data objects:

$$I = \{I_{\text{doc}}^{\text{in}}\} \cup \{I_{\text{dat}}^{\text{in}}\} \cup \{I_{\text{doc}}^{\text{out}}\} \cup \{I_{\text{dat}}^{\text{out}}\} \quad (2)$$

The workflow parameters $\{P_{i, i=1,6}\}$ cover: $P_1 = T_{\text{ex}}$ – execution time; $P_2 = R$ – resource, required for execution; $P_3 = A$ – ability to be automatically executed; $P_4 = S$ – set of subsystems, used for workflow execution; $P_5 = F^S$ – set of separate subsystem functions, realizing task execution; $P_6 = P_{\text{ad}}$ – set of additional workflow parameters.

Separate work models can be represented in a formal manner as:

$$O = (O, I^O, P) \quad (3)$$

where O is the set of identification objects; I^O is the set of informational objects; P is the set of service provision parameters.

Identification objects $\{O_{id, id=1,3}\}$ include: $O_1 = N_o$ – name; $O_2 = d$ – description; $O_3 = E$ – executor. Task informational objects include income and outcome informational objects (document and data). Set of work parameters $\{P_{i, i=1,7}^O\}$ include: $P_{O_1}^O = \xi_{kl}(r_{kl})$ – execution time of work l of stage k ; $P_{O_2}^O = r_{kl}$ – resource, required for execution work l of stage k ; $P_{O_3}^O = a$ – the ability to be automatically executed; $P_{O_4}^O = S$ – set of OSS/BSS¹² subsystems, used for workflow execution; $P_{O_5}^O = F^S$ – set of OSS/BSS subsystem functions, realizing works execution; $P_{O_6}^O = R_O^n$ – work realization alternatives, defining execution time and resource values:

$$R_O^n = (N_R, \xi_{kl}^n(r_{kl}), r_{kl}^n) \quad (4)$$

$P_{O_7}^O = P_{\text{ad}}$ – set of additional work parameters.

The proposed model allows to perform workflow analysis at a planning stage, applying graph theory and optimization algorithm, represented below.

Workflow Analysis

The workflow analysis method, providing workflow model verification, execution time minimization and automating its transformation, is briefly described below.

The workflow graph model can be represented as sequential stages, containing a few parallel executed tasks, enabling the definition of execution time as follows:

$$T_{\text{ex}} = \sum_k \max_l \xi_{kl}(r_{kl}) \quad (5)$$

The suggested method of workflow analysis is represented mathematically as:

$$M = (G_{\text{f}}, G_{\text{v}}, M_{\text{min}}, M_{\text{tr}}) \quad (6)$$

¹² Operation Support System/Business Support System.

where G_f is the graph generating procedure; G_v is the graph connectivity verification; M_{min} is the execution time minimization and M_{tr} is the diagram into realization model transformation. Each of the procedure is described in details in [6].

Each work of the workflow has one to three implementation variants, defining execution time and resources, the task is to find such implementation variant for each task to minimize the total workflow execution time, when the total resource is limited and known. The objective function of the task is represented below:

$$F(r) = \min_{\sum_{kl} r_{kl} = r} M \sum_k \max_l \xi_{kl}(r_{kl}) \tag{7}$$

It is also necessary to find the implementation variant for each task, i.e. $\{r_{kl}, k = 1, \dots, n, l = 1, \dots, m\}$, where the required minimum of time (7) is reached. Solution for function (7), applying dynamic programming, is described in [6].

3.2 NFR Balancing Method

The proposed NFR balancing method is based on creating FR and NFR collaboration model. Implementation of functional requirements is presented by listed FB¹³. Each of FB is responsible for particular logical function. The proposed method includes NFR Catalogue development, FR decomposition, NFR mapping, FB distribution and Balancing stages. NFR balancing method uses NFR Catalogue and FR and create collaboration model between them. The main stages of the concept are described in [16] and briefly represented below.

Catalogue of NFR. It is proposed to operate with catalogues for performance and serviceability. Catalogues are updated with further operationalizations to keep own NFR catalogues updated, which facilitates reuse of acquired knowledge on NFR.

FR decomposition. It describes influence of services features on NFR. Services and features are depicted for each FB (Table 1).

Table 1. FR decomposition

Service	Functional Block	Functional Requirement
Service1	FB1.1 or FB1.2	FR1, FR2
Service1	FB2.1 and FB2.2	FR3, FR1
Service2	FB1.1	FR5, FR6
Service2	FB3	FR1, FR7

NFR mapping. Each call of FB requests a defined amount of system resources (memory, processor time, etc.) and has list of characteristics: response time, availability, etc. These characteristics are mapped to NFR from catalogue with values, specifying how exact FB meets particular NFR (it is graded from 0 to 100, Table 2). One FB can provide the same functionality with different NFR (FB1.1, FB1.2). From functional point of view there is no difference between these two blocks. The difference is

¹³ Functional block.

how each FB meets the NFR. NFR mapping requires considering every service and connecting it to required FB, covering requested functionality.

Table 2. NFR mapping

Functional block/ NFR	Availability	Performance	Security
FB1.1	90	80	10
FB1.2	80	70	20
FB2.1	50	10	10
FB2.2	5	20	30

FB distribution. Using NFR catalogues and FR decomposition, FB distribution can be realized. Fig.1 shows that FR 1 and FR 2 from Table 1 can be implemented either by FB1.1 or FB1.2. The implementation way depends on NFR specification for particular case.

Balancing and target models. The target model is obtained by using balancing between NFR and approaches to implementation of a particular functionality with FB. For instance, there is the Customer’s demand for service supporting the highest availability and there is no specified requirement for security and performance. Such case can be realized by the model, represented on Fig.2. A priority is assigned to the requirements, considered during target model development.

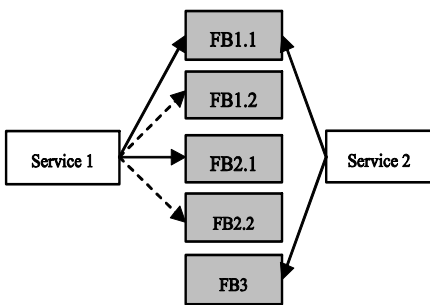


Fig. 1. Functional blocks distribution

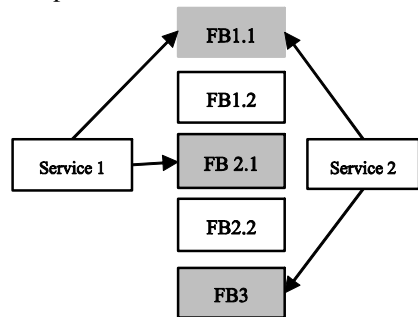


Fig. 2. Target deployment model

3.3 Modified Service Workflows Transformation Method

The modified method is based on BPD2BPEL translation algorithm [17] which can translate arbitrary diagrams into BPEL code. BPMN extension (BPNE) [5] has been taken as an input extended business process diagram.

Key feature of proposed method is modified analysis process. When the element is analyzed, its additional parameters are also checked. If new parameters are found, information about them is saved for their subsequent transformation into BPEL elements. Proposed method considers each BPNE element.

4 Service Design Tool

BPMA. The proposed workflow design method is implemented in a new developed tool BPMA¹⁴ (fig.3). It uses standard GTK+, Dia diagram editor, PyDia interface, Python Interpreter, PyGTK¹⁵ and developed module BPEA, implementing workflow modeling and analysis algorithms. Main BPMA components functions are described further. “Init” realizes Dia and user intercommunication and launches modules functions. “Props” provides setting, changing and saving of workflows and its objects parameters. “Bplyzer” implements time minimization algorithms. “Transform” implements the transformation logic to create a realization diagram from workflow requirements. “Reports” generates reports on workflow modeling and analysis.

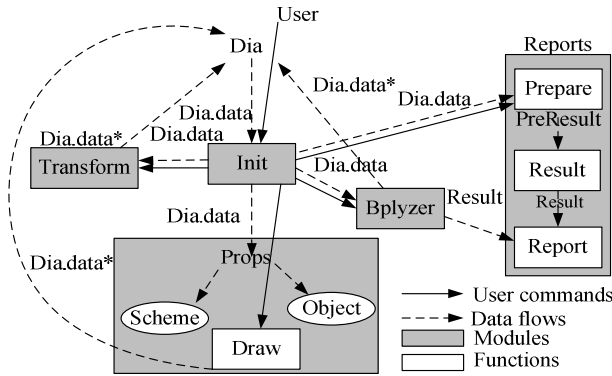


Fig. 3. BPMA functioning scheme

BPNE-BPEL transformation tool implements presented BPNE-BPEL transformation method (fig.4) [18].

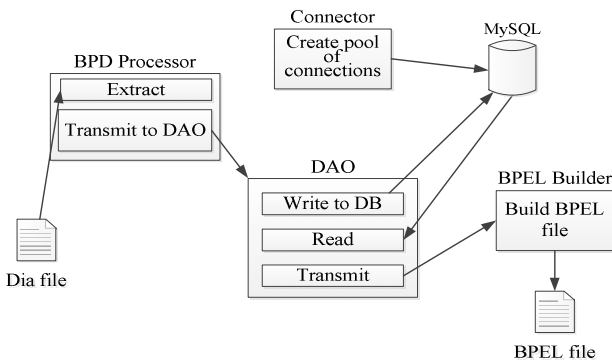


Fig. 4. Structural scheme of the BPNE-BPEL transformation tool

¹⁴ Business Process Modeling & Analysis.
¹⁵ Set of Python wrappers for the GTK+ GUI library.

The tool performs transformation of data, extracted from BPNE file, into representation form, convenient for the subsequent work; translation of BPNE elements into BPEL code, using its object-oriented model; gathering all translated elements and components in a single BPEL file.

4.1 Case-Study

Workflow design. BPMA has been tested during the planning and design of several services, such as «Bonus program», «Selling Electronic Voucher», «Tariff Plan Change» [6], in the SITRONICS Telecom Solutions company. Testing results have shown that for services, having 1-1.5 months of development time, it can be reduced by 3-5 days and development costs can be decreased by 5-7 man-days for one service; the proposed workflow analysis can reduce the time used for service provision up to 20 seconds for the services, provided in 3-5 minutes by finding the combination of tasks implementation variants, having the same resource limit.

NFR balancing. Evaluation of the proposed NFR balancing method is demonstrated on charging of GPRS service. Its FR decomposition is shown in Table 3. GPRS service availability and service delay are taken as main NFRs. Considering statistical data and enterprise knowledgebase, GPRS service characteristics were estimated (Table 4) and the GPRS service target model (fig.5) was got. The target model provides service optimal deployment configuration, ensuring service highest availability with minimal delay. The proposed method is experimentally proved to be efficient when services design and deployment, allowing NFR modeling, providing service functionality with required functional and non-functional parameters.

Table 3. FR decomposition of GPRS service

Service	Functional Block	Functional Requirement
GPRS	LBS1.1 or LBS1.2	Location Base Charging
GPRS	RF2.1 and RF2.2	Step Charging
GPRS	NB3.1 or NB 3.2 or NB3.3.	User notification

Table 4. NFR mapping of GPRS service

Functional block/ NFR	Availability	Delay
LBS1.1- location based module implemented as internal cache in Online Charging System	90	80
LBS1.2 –external Home Zone Billing -HZB platform	50	10
RF2.1 – internal Rating	50	20
RF2.2 – external Rating	5	15
NB3.1 – notification via SMS	40	50
NB3.2 –online notification via USSD	50	40
NB3.3 – offline notification via email	50	10

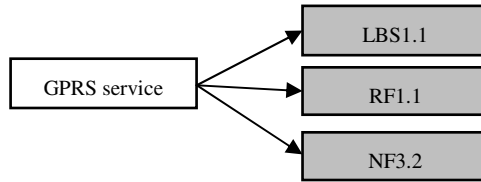


Fig. 5. Target model for GPRS service

5 Summary and Outlook

A novel approach for analyzing and developing modern services, provided by web-oriented applications, has been proposed. The discussion is focused on the development of a computational independent workflow model, its analysis, transformation of abstract into executional workflows, and also aspect of NFR balancing when service development. The proposed workflow model provides formalization of service provision parameters, required for the planning stage. The analysis methods allow automating service planning and minimizing service provision time. The workflow design method reduces the time of service re-engineering up to 10% by improving and automating the service planning stage.

The proposed NFR balancing method provides NFR description, considering collaboration between services and meeting NFR by system FB. The method increases efficiency of service development on design and deployment stages and allows fast system reconfiguration on customer demand.

Modified BPMN-BPEL transformation method and tool provide automation of abstract workflows transformation into execution ones, allowing to consider workflow required parameters.

Future work will focus on analysis of computing workflows and their enactment, services monitoring and re-configuration, which are strongly connected to the design and analysis stages. NFR balancing method will be extended with possibility to consider possible change of NFR list and their priorities during different time periods (e.g. periods with high load) and also take into account changing priority between services.

Key aspects of the proposed approach to future service development can be applied to extend ISO/IEC/IEEE 29148:2011 Standard, dealing with the processes and products related to the engineering of requirements for systems and software products and services throughout the life cycle, to raise the overall efficiency of service development and re-engineering.

References

1. Terplan, K.: OSS Essentials: Support System Solutions for Service Providers, 610 p. John Wiley, New York (2001)
2. RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, <http://tools.ietf.org/html/rfc2474>

3. Recker, J., Mendling, J.: On the Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In: CAiSE 2006 Workshop Proceedings - Eleventh International Workshop on EMMSAD, Luxembourg, June 5-6, pp. 521–532 (2006)
4. Börger, E.: Approaches to modeling business processes: A critical analysis of BPMN, workflow patterns and YAWL. *Software and System Modeling* 11(3), 305–318 (2011)
5. Kot, T., Globa, L., Schill, A.: Applying business process modeling method when telecommunication services development. In: Proceedings of International Conference on Microwave and Telecommunication Technology (CriMiCo), Sevastopol, vol. 1, pp. 457–458 (2011)
6. Kot, T., Reverchuk, A., Globa, L., Schill, A.: A novel approach to increase efficiency of OSS/BSS workflow planning and design. In: Abramowicz, W., Kriksciuniene, D., Sakalauskas, V. (eds.) BIS 2012. LNBIP, vol. 117, pp. 142–152. Springer, Heidelberg (2012)
7. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: *Workflows for e-Science*. Scientific Workflows for Grids, 523 p. Springer (2007)
8. van der Aalst, W.: BPM and Workflow Analysis. *BPTrends* 5(4), 1–2 (2007)
9. Software Errors Cost U.S. Economy \$59.5 Billion Annually. NIST Assesses Technical Needs of Industry to Improve Software-Testing (2002), <http://www.cse.buffalo.edu/~mikeb/Billions.pdf>
10. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: *Non-Functional Requirements in Software Engineering*, 472 p. Kluwer Academic Publishers, Boston (2000)
11. A KAOS Tutorial (2007), <http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>
12. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: *Social Modeling for Requirements Engineering*, 752 p. MIT Press, Cambridge (2011)
13. van Lamsweerde, A.: *Requirements Engineering: From System Goals to UML Models to Software Specifications*, 650 p. John Wiley and Sons (2009)
14. Mendling, J., Lassen, K., Zdun, U.: Transformation Strategies between Block-Oriented and Graph-Oriented Process Modelling Languages. In: MKWI 2006, Band 2, XML4BPM Track, pp. 297–312. GITO-Verlag, Berlin (2006) ISBN 3-936771-62-6
15. Mendling, J., Lassen, K., Zdun, U.: On the Transformation of Control Flow between Block-Oriented and Graph-Oriented Process Modeling Languages. *IJBPM*. Special Issue on Model-Driven Engineering of Executable Business Process Models 3(2), 96–108 (2008)
16. Kot, T., Reverchuk, A., Globa, L., Schill, A.: Method of non-functional requirements balancing when service development. In: Proceedings of the International Multi-Conference ACS-AISBIS 2012 (May-June 2012), Szczecin: Journal of Theoretical and Applied Computer Science 6(3), 50–57 (2012)
17. Ouyang, C., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Mendling, J.: From Business Process Models to Process-Oriented Software Systems. *ACM Transactions on Software Engineering and Methodology* 19, 1–37 (2006)
18. Pukhkaiev, D., Kot, T., Globa, L.: Translating BPMN/BPNE into BPEL. In: Proceedings of 11th International Conference TCSET 2012, Lviv-Slavske, Ukraine, vol. 1, p. 386 (2012)