UDC 621.39

COMPLEX APPROACH TO SERVICE DEVELOPMENT

Tetiana M. Kot[1], Andrey V. Reverchuk[1], Larysa S. Globa[1], Alexander Schill[2]

[1]National Technical University of Ukraine "Kyiv Polytechnic Institute", Kyiv, Ukraine,
[2]Dresden University of Technology, Dresden, Germany

Modern companies including telecommunication companies and mobile operators working in the global environment should guarantee technological effectiveness and innovation, renewing their technologies and services. Operation Support System/Business Support System is used in telecommunication companies. In current state-of-the-art approaches, several iterations involving analysts and system architects are necessary, methodologies allow modeling non-functional or functional requirements but they do not take into account the interaction between functional and non-functional requirements as well as collaboration between services. Web Services Agreement is a convenient way to contain QoS parameters but state-of-the-art SLA-aware methods cannot support all classes of non-functional parameters and provide run-time support and dynamic reconfiguration at the same time. 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 including support of all classes of non-functional parameters and providing run-time support and dynamic reconfiguration of provided services.

Introduction

Today, companies working in the global environment constantly renovate and improve their technologies used Web-oriented applications and provided services. This is implemented via service development and reengineering. 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 as well as 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: development and description of independent and computational workflows including service planning, applications prototyping and creating several possible realization models; workflow analysis and simulation considering several system parameters and modeling its behavior; workflow enactment providing deployment models considered both hardware and software service deployment alternatives; service/workflows monitoring leading to services reengineering and/or reconfiguration.

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. Furthermore, in the current state-of-the-art computational independent workflow design performed by using the existing notations and tools on the one hand does not consider all required parameters necessary on the service planning stage such as numerical values of execution time and resources required for that as well as the document and information flows supporting service provision, and on the other hand provides poor connection to system functionality which should realize these workflows.

Because the service design and deployment should be considered concern non-functional requirements (NFR) to service provision, it is extremely important to meet NFR for real time systems specifying "quality characteristics" or "quality attributes". Generally NFR could be classified as follows: Performance (Response Time, Throughput, Utilization, Static Volumetric); Scalability; Capacity; Availability; Reliability; Recoverability; Maintainability; Serviceability; Security; Regulatory; Manageability; Environmental; Data Integrity; Usability; Interoperability. If NFR are not considered on the design stage, the provided service may be useless in practice.

Currently, NFR are not considered with perspective of provided services list as legacy methods can design the 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. Balancing allows modeling the system behavior for requested list of services to analyze how this configuration meets the NFR.

The transformation of processes between workflow modeling stages is still an open question. The existing transformation standards and languages that can be applied are not fully automated and have to be investigated to make service design consistent and correct.

ISSN 2219-9454, Telecommunication Sciences, 2012, Volume 3, Number 2

One of the most considerable advantages of Web services technology is the possibility of connecting them together to implement the high level business process. Web Service Composition (WSC) is a method that allows performing such connections. Fast changes of requirements need dynamic composition and reconfiguration of services considering required values of functional and non-functional parameters, which represent Quality of Service (QoS) characteristics such as reliability, availability, response time, cost, performance. Hence, Web Service Composition unites stages of Process enactment and Diagnosis of BPM life cycle. One problem of WSC is that the QoS parameters should be taken into account on stages of Process enactment and Diagnosis, and the most convenient way to contain and monitor their values is applying Service Level Agreement (SLA). Another problem of WSC is related to the evaluation of QoS parameters. Despite the existence of various dynamic composition approaches, there are no techniques combining evaluation of different classes of QoS parameters during performing composition and none of them can meet all QoS characteristics described further.

This paper describes an NFR approach to service development including novel computational independent workflow design method, NFR balancing method, modified service workflows transformation method and SLA-aware method of WSC whereas they allow taking into consideration different QoS parameters.

In the paper, the following aspects of service development are considered below. The state-of-the-art analysis of workflow design notations and tools as well as state-of-the-art analysis of methods and approaches considering NFR are reflected. NFR analysis methods and service implementation technologies are described. BPMN to BPEL (Business Process Execution Language) transformation strategies and Web-service composition approaches are also considered.

NFR approach to service development is introduced. It includes: the workflow design method focusing on a computational independent workflow model describing service provision and considering service provision parameters required for their analysis; the workflow analysis method providing minimization of service provision time; NFR balancing method focusing on collaboration of functional and non-functional requirements; modified service workflows transformation method; SLA-aware method of Web-service composition.

A prototypical realization of the advanced approach including design and transformation tools as well as the concept of web-service composition module and highlights evaluation results of the developed tools are presented. The evaluation using a real-world scenario within a telecommunication company is demonstrated. A summary and outlook on future work are concluded in the paper.

State of the art and background

Workflow design notations and tools

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

Computational independent workflow design

Computational independent workflows are designed using graphical standards and allowing their formalization and their possible flows and transitions in a diagrammatic way. The analysis has shown that in practice computational independent workflows are usually designed using graphical notations such as BPMN 2.0, UML AD (UML Activity Diagram), USLD and tools such as CA ERwin Process Modeler, Enterprise Architect and MS Visio.

USDL sufficiently generalized under development stage does not fully meet all the requirements of workflows analysis and design. Additionally, its usage is difficult due to its complexity in spite of its comprehensiveness.

BPMN 2.0 shortcomings are clearly described in [1]. The central argument against using regular BPMN is that resource management can be expressed only via lanes (actors, roles) or performers of user or manual tasks. No execution time parameters are considered. All further existing workflow modeling notations retain this criticism in general.

Nevertheless, BPMN providing the ability of computational independence to computing workflows transformation (BPEL diagrams) widespread in industry can be applied as a basic notation for computational independent workflow design. Thus, it is intended to extend it by adding the missing concepts.

Workflow analysis

The given short overview of the workflow analysis methods and tools has shown that there are two types of analysis both considering computational workflow. Monte Carlo design time simulations as well as Petri Nets analysis can be applied to Petri Nets with their further analysis employing transformation approaches for BPMN, UML AD, EPC and BPEL. USLD diagrams can be analyzed using service ontological analysis. The runtime analysis can be employed in order to define, for instance, the process mining based on the execution logs.

The software such as Pegasus, Cactus, ASKALON, GLUE [2] can be used for analysis of these fields. Mentioned and analyzed current possibilities for this task stage are very limited. Shortcomings of workflow analysis methods and tools are extensively described in [3]. The central criticism is that the stage of requirements analysis is implemented principally in a manual manner.

Service planning

In differentiated service models, the service provision depending on content and communication technologies is characterized by parameters such as Quality of Service, acceptable service delay as well as the capacity depending on radio technology. Besides, separate tasks of the workflow can have a few variants of implementation [3]. Their values determine service provision time and resources which are to be defined at the planning stage.

Thus, workflow tasks can be implemented in different ways. The implementation variant is defined by time and resources required for service provision. The total resource required for service provision can be presented as a sum of resources required to implement each task of the workflow:

$$R = \sum_{k,l,n} r_{kl}^{(n)} , \quad (1)$$

where $n = \overline{1,N}$; $k = \overline{1,K}$, $K \le 5$; N denotes the number of workflow tasks describing the service provision; K is the number of implementation variants for the workflow tasks.

When applying the service planning, operators work with the following data: total amount of resources R required for service provision; resource $r_{kl}^{(n)}$ required for implementation of task l on stage k using realization variant n; execution time $\xi_{kl}^{(n)}$ of task l on stage k using realization variant n.

The time dependence versus the resource $\xi(r)$ is not linear function for all tasks of implemented variants. This circumstance should be considered at the planning stage during the computational independent workflow design.

Methods and approaches to considering NFR

Early-phase requirements engineering should address organizational and non-functional requirements, while later-phase engineering focuses on completeness, consistency and automated verification of requirements. There are reports [4] showing that not proper dealing with NFR has led to considerable delays in the project and consequently to significant increase of the final costs. One of the most important reasons of such delays is neglecting performance during software development leading to several changes in both hardware and software architectures as well as in software design and code. The system can be deactivated just after its deployment because of neglecting such NFR as reliability, cost, usability and performance.

NFR are considered on design stage and there are several approaches to model NFR within the scope of developed service. NFR framework [5] is a methodology that guides the system to accommodate change with replaceable components. NFR framework is a goal-oriented and process-oriented approach guiding the NFR modeling. NFR as security, accuracy, performance and cost are used to drive the overall design process and choosing design alternatives. It helps developers express NFR explicitly, deal with them systematically and use them to drive development process rationally.

KAOS [6] is another methodology considering NFR. It allows requirements engineering enabling analysts to build requirements models and derive requirements documents from KAOS models. KAOS is independent of the development model type.

There are some other approaches and methods to NFR modeling [7], but all of them don’t consider collaboration between functional requirements (FR) and NFR. The legacy software tools (NFR-Assistant CASE, ARIS) don’t provide functionality to model NFR and consider their influence on system functionality.

BPMN to BPEL transformation strategies

There are various BPMN to BPEL transformation strategies such as Element-Preservation, Element-Minimization, Structure-Maximization and Structure-Identification, described in [8], Event-Condition-Action-Rules and others. A basic idea of mapping used as a core of most algorithms for translating BPMN or any other WF-net into BPEL code consists in applying a mix of mentioned strategies.

Two categories of tools are applied in practice. BPMN graph is serialized to an XML document. After that the XML document is translated into an abstract

BPEL document in an automatic way. The abstract BPEL is enriched with pieces of information necessary to make it executable. BPMN graph is translated directly into executable BPEL code. This is possible when input/output files of the future Web Service (WSDL files) are created in advance.

There are three main problems of BPMN to BPEL translation. There is fundamental mismatch between these two languages. BPMN is a graph-structured notation while BPEL is block-structured language. Readability of the resulting BPEL code is very low. Translation of extended notations considering additional workflow parameters is not possible. Both BPMN and BPEL are declared to be extensible. However, most vendors ignore this ability. Hence, development of the method providing automatic translation of extended BPMN elements is an important aspect in service development process.

Web-service composition approaches

QoS parameters became extremely important as mentioned above. An overview of several approaches for modeling web service quality composition is represented below.

The approach to estimate workflow execution time and cost based on continuous-time Markov chains is proposed in [9]. The simple QoS model is provided via assigning an execution cost to each activity, whose start and end are signalled by the service. When the activity is started, the labelled sum is deducted. To this end, each transition is labelled with the sum of costs of all activities being active in the destination node.

Description of elementary service quality as a quality vector is proposed in [10], a similar solution can be found in [11]. The authors propose to calculate the vector components of quality criteria for composite services by using special aggregation functions.

Global planning approach [12] is used to select optimal components of the composite service. The service selection can be formulated as an optimization problem which can be solved by using the linear programming.

The use of agent-oriented Tropos methodology to model web service properties is proposed in [13]. To provide modelling of a quality composition, interacting services are represented as a multi-agent system.

An ontology-based framework for dynamic web services selection is presented in [14]. The base is two-layered model. The first layer is service ontology, whereas the second layer is QoS ontology. The QoS correspondence is defined in the services ontology when the QoS ontology describes the quality concepts. The QoS ontology is separated into lower, middle and upper ontologies. QoS upper ontology includes basic characteristics of all qualities and the main concepts associated with them. QoS middle ontology specifies domain-independent quality concepts such as availability, performance, reliability, security and is typically completed by domain-specific QoS lower ontology.

It should be noted that Ontology-based solution relies on ontologies only as knowledge base and for decision making operations. In fact, it utilizes the agent-based architecture. Nevertheless, it is important to distinguish this approach from other agent-oriented approaches because ontologies can focus not only on Objective, but also on Subjective QoS parameters. At the same time, it lacks runtime support and QoS assignments.

Another important issue is that this method is presented as the dynamic selection approach based on QoS. However, it can be easily used as WSC approach where the only difference is that services are selected not for user directly, but for an application which creates a sequence of chosen services.

Table 1 summarizes the results of comparison of WSC approaches presented earlier with respect to QoS characteristics such as Objective and Subjective QoS, Run-time support, QoS assignment to composite service, Quality Requirements.

Table 1. QoS-based comparison of WSC approaches.

Table requirements	WSC Approaches			
	Markov Chains	Quality Vector	Agent-oriented	Ontology-based
Objective QoS	+	+	+	+
Subjective QoS	–	–	–	+
Run-time support	+	–	+	–
QoS assignment	–	+	+	–
Requirements considering level	Low	Average	High	High

Approaches of WSC should maintain the run-time support in order to provide monitoring QoS parameters and respond to their changes. Markov chains- and agents-based approaches [9], [13] provide the run-time support. When quality constraints and preferences are assigned to composite services, it is easy to reuse these composite services in another quality driven composition. Table 1 shows that quality vector-based and agent-oriented solutions allow assigning QoS to composite service.

QoS requirements for Web Services include performance, reliability, interoperability, exception handling, robustness, scalability, capacity, accessibility, accuracy, integrity, availability, security, and network-related QoS requirements. Not all the requirements are met in approaches presented above. Specifically, execution price and time are estimated in [9]. While [10] considers execution price, execution duration, reliability, availability and reputation, [11] focuses on four quality dimensions such as execution cost, execution time, reliability, and availability. Ontology- and agent-based solutions theoretically deal with most of QoS requirements.

The analysis of data presented in Table 1 shows that the current state-of-the-art QoS models of WSC is not ideal at all. None of the presented approaches can meet all QoS characteristics. The most crucial characteristic is the ability to support all types of QoS parameters. Markov chains solution and Quality Vector solution show weak results regarding this characteristic while Ontology-based and agent-based solutions display good results relative to some types of QoS parameters including non-measurable parameters that should be taken into account.

SLA enables a convenient way to perform WSC and monitoring the composed service. Applying SLA to methods mentioned above provides much more acceptable results than those presented in Table 1. For instance, run-time support is now available for all existing methods using SLA as shown in Table 2.

Table 2. QoS-based SLA-aware comparison of WSC approaches.

Requirements	WSC Approaches			
	Markov Chains	Quality Vector	Agent-oriented	Ontology-based
Objective QoS	+	+	+	+
Subjective QoS	–	–	–	+
Run-time support	+	+	+	+
QoS assignment	+	+	+	+
Requirements considering level	Low	Average	High	High

Table 2 shows that applying SLA to WSC methods provides Run-time support and QoS assignment for all approaches. Wherein, it is assumed that WS-Agreement for composite service is generated by using WS-Agreements of single services. As follows from Table 2, the Ontology-based WSC approach in combination with SLA-awareness allows performing the most reliable WSC of all approaches presented earlier.

NFR approach to service development

The proposed advanced approach to future service development includes: novel computational independent workflow design method based on the workflow model and its analysis capable providing automate service planning stages; NFR balancing method focusing on collaboration between functional and non-functional requirements to service planning; modified service workflows transformation method providing the abstract to execution workflows transformation considering workflows required parameters. Each of the proposed methods is briefly described below.

Workflow design method

The workflow design includes its modelling and simulation. The novel method of workflow design presented in this section is focused on computational independent workflow. It consists of extended modelling the computational independent workflow and its analysis, which include forming the workflow graph and verifying its connectivity, execution time minimization under transformation of workflow to realization diagram.

The suggested method modifies Model Driven Architecture (MDA) on the business logic level as shown in Fig. 1. The main features of proposed method are extensively discussed in [3].

Workflow model

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

The mathematical formalization of the workflow can be expressed by the relation:

$$BP = (E, I, P), \qquad (2)$$

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

The identification objects are described by $\{E_{id, id=1,4}\}$, where E_1, E_2, E_3 denote name, description, executor, respectively; $E_4 = O$ is the set of works.

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

$$I = \{I_{doc}^{(in)}\} \cup \{I_{dat}^{(in)}\} \cup \{I_{doc}^{(out)}\} \cup \{I_{dat}^{(out)}\}. \quad (3)$$

The workflow parameters are described by $\{P_{i,i=1,6}\}$, where $P_1 = T_{ex}$ denotes the workflow execution time; $P_2 = R$ defines the resource required for workflow execution; $P_3 = A$ specifies the ability to be automatically executed; $P_4 = S$ is the set of OSS/BSS subsystems used for workflow execution; $P_5 = F^{(S)}$ is the set of OSS/BSS separate subsystem functions realizing the task execution; $P_6 = P_{ad}$ is the set of additional workflow parameters.

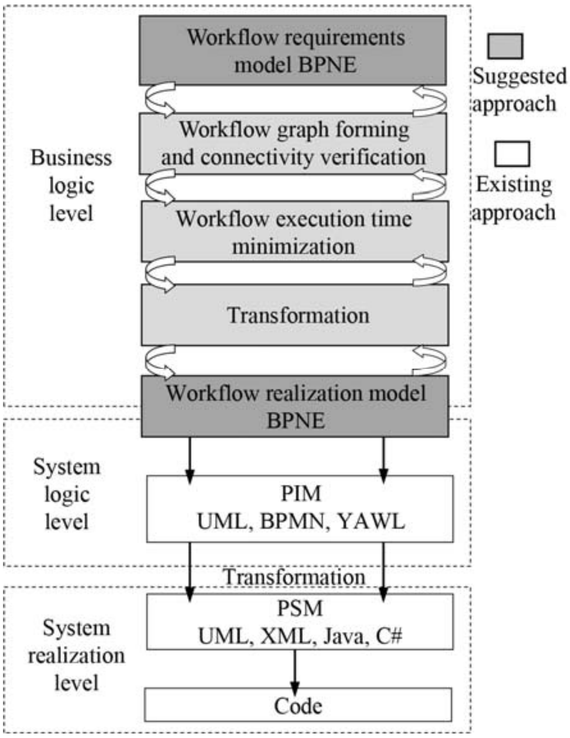

Fig. 1. Workflow design method extending the OMG MDA approach.

Separate work models can be presented in a formal manner as follows:

$$O = [O, I^{(O)}, P], \quad (4)$$

where O defines the set of identification objects; $I^{(O)}$ denotes the set of informational objects; P is the set of parameters characterizing service provision concerning separate work realization.

The set of identification objects is described by $\{O_{id,i=1,3}\}$, where $O_1 = N_O$, $O_2 = d$, $O_3 = E$ denote name, description, executor, respectively. The income and outcome informational objects include documents and data objects.

The set of work parameters can be described as $\{P_{i,i=1,7}^{(O)}\}$, where $P_1^{(O)} = \xi_{kl}(r_{kl})$ denotes the execution time of the work l on the stage k; $P_2^{(O)} = r_{kl}$ defines the resource required for execution of the work l on the stage k; $P_3^{(O)} = a$ specifies the ability to be automatically executed; $P_4^{(O)} = S$ is the set of OSS/BSS subsystems used for workflow execution; $P_5^{(O)} = F^{(S)}$ is the set of OSS/BSS separate subsystem functions realizing works execution; $P_6^{(O)} = R_O^{(n)}$ denotes work realization alternatives defining execution time and resource values; $P_7^{(O)} = P_{ad}$ is the set of additional work parameters. The work realization alternatives can be formally described by the relation

$$R_O^{(n)} = [N_R, \xi_{kl}^{(n)}(r_{kl}), r_{kl}^{(n)}]. \quad (5)$$

The presented model is extensively described in [15]. It allows performing the workflow analysis at a planning stage with applying graph theory and optimization algorithm presented below.

Workflow analysis

The workflow analysis approach is the second central point of the proposed design method. The workflow graph model shown in Fig. 2 can be represented as sequential stages containing a few parallel executed tasks or just one task enabling the definition of workflow execution time as follows:

$$T_{ex} = \sum_k^n t_{st} = \sum_k^n \max_l \xi_{kl}(r_{kl}). \quad (6)$$

The suggested method of workflow analysis providing workflow model verification, execution time minimization and automating its transformation can be mathematically described by the following relation:

$$M = (G_f, G_V, M_{min}, M_{tr}), \quad (7)$$

where G_f denotes the graph generating procedure; G_V designates the graph connectivity verification; M_{min} defines the execution time minimization; M_{tr} denotes the diagram representing the model of realized transformation.

According to the proposed method, workflow analysis is performed as follows. When the workflow diagram is designed having several implementation variants of its works, the workflow graph is generated and its connectivity and syntax are verified. The execution time of the verified workflow model can be found in the next step having the general resource value as a limitation. When the time minimization procedure is realized, only one variant implemented for each operation can be selected. The workflow model with minimal execution time is transformed into the realization model describing the system modules, their functions, requirements to time and resources. Each of the procedures is described in details in [3].

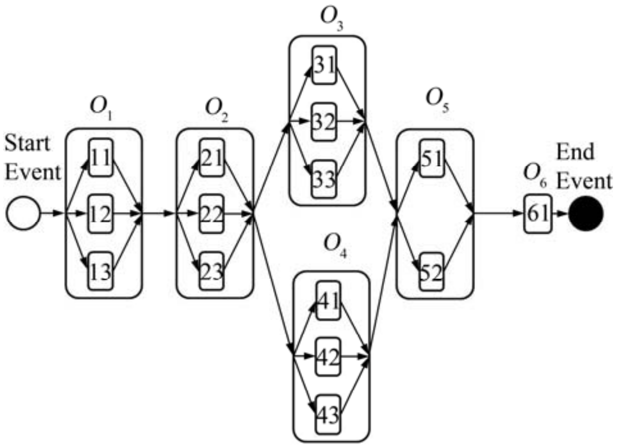

Fig. 2. Workflow graph with task implementation alternatives: (11) task 1 execution alternative having ξ_{11} time and consuming r_{11} resources; (12) task 1 execution alternative having ξ_{12} time and consuming r_{12} resources; (13) task 1 execution alternative having ξ_{13} time and consuming r_{13} resources; (21) task 2 execution alternative having ξ_{12} time and consuming r_{21} resources; and so on.

The task of finding the minimal workflow execution time is *P*-complete and can be solved in polynomial time. The objective function of the task can be represented in following form:

$$F(r) = \min_{\sum_{kl} r_{kl} = r} M \sum_k \max_l \xi_{kl}(r_{kl}) . \quad (8)$$

According to (8), the implementation variant r_{kl} ($k = 1,...,n$; $l = 1,...,m$) can be found for each task, where the required minimum of time is reached. To find the function specified in (8), dynamic programming is applied.

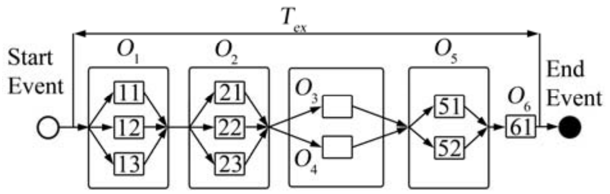

Fig. 3. Workflow time minimization.

NFR balancing method

The proposed NFR balancing method is based on collaboration of FR and NFR models. Implementation of functional requirements is presented by listed functional block (FB). Each FB is responsible for particular logical function. The proposed method includes the following main stages: NFR catalogue development; FR decomposition; NFR mapping; FB distribution; balancing; target deployment model.

NFR balancing method uses NFR Catalogue, Functional Requirements to be implemented, created collaboration model between them. The main stages of the concept are described in [15] and briefly considered below.

The proposed NFR catalogue is destined to operate with catalogues for performance and serviceability. These catalogues are updated with further operations to create own innovated NFR catalogues. This facilitates future reuse of acquired knowledge on NFR elicitation.

FR decomposition describes the influence of service features on NFR. Services and features are depicted for each functional block. Example of FB distribution between services is presented in Table 3.

Table 3. 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

Each call of FB requests a defined amount of each system resource (memory, processor time) and has a list of characteristics such as response time, availability. These characteristics shall be mapped to NFR from catalogue with values specifying how exact FB meets the particular NFR. It could be graded from 0 to 100 as shown in Table 4. 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 only how each FB meets the NFR. NFR modeling requires considering every service and connecting it to required FB covering requested functionality.

Table 4. 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

Using NFR catalogues and FR decomposition, functional blocks distribution can be realized. Fig. 4 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 a particular case.

The target model would be obtained by using the balance 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 are no specified requirements for security and performance.

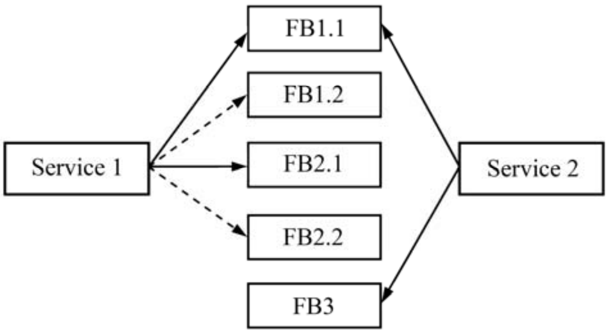

Fig. 4. Functional blocks distribution

Such case can be realized by the model represented in Fig. 5. A priority should be assigned to any requirement considered during target model development.

Modified service workflows transformation method

The modified method is based on BPD2BPEL translation algorithm which can transform the arbitrary diagram into BPEL code. BPMN extension such as BPNE [16] has been taken as an input extended business process diagram.

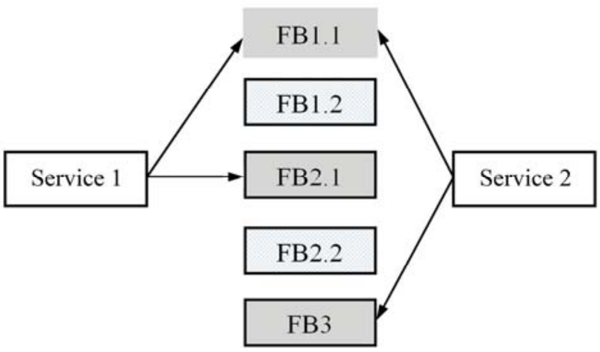

Fig. 5. Target deployment model.

The 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 on them is saved for their subsequent transformation into BPEL elements. The proposed method considers each BPNE element.

SLA-aware method of Web-service composition

Ontology-based approach implemented agent-based architecture is realized in the Web Services Agent Framework (WSAF) [14]. WSAF includes service selection agents that use the QoS ontology and an XML policy language that allows service consumers and providers to expose their quality preferences and advertisements.

Agent for communication with service is called when a consumer application built with WSAF desires to use a service. For the each service a service agent is created to expose the service interface enlarged with functionality to capture the consumer's QoS preferences (policies) and provide agencies or other agents query for a suitable match. The service agent can determine the values of objective QoS-attributes (reliability, availability, request-to-response time) and get user feedback for subjective attributes such as the user's overall experience. Afterwards values of these QoS are transferred to the appropriate agencies.

A typical consumer-to-agent interaction and control flow is described below. Upon initialization, WSAF sets up all configured agencies [14]. Providers register service implementations with WSAF by configuring each service in terms of WSDL URIs, service domains, and the service's advertised QoS policy. Each configured service interface has an agent. The consumer application creates a local proxy object for the service agent (the consumer invokes the proxy with its policy). The agent uses the policy and its configuration to load and to run its script. The script typically consults the QoS and service ontologies to complete its configuration. This setup occurs once per consumer-to-agent interaction episode. By default, the agent performs a binding operation once configured. Consumers can initiate a rebinding or specify an automatic rebinding initiation in their policies. The agent selects a service implementation based on agency data, and then dynamically creates a proxy object for each selected service. The consumer invokes the agent's service operations. Each invocation is forwarded to the service proxy while being monitored by the agent. When the service responds, the agent inserts appropriate data to the relevant agencies.

The service agent finds services matching the given interface using UDDI. After this, it applies the consumer's policy on the available quality data providing service implementations ranking. The ranking is calculated according to quality degree match based on the provider's reputation for the given quality.

Applying SLA to this model allows the system selection based on its rankings and evaluating parameters stored in WS-Agreement. Moreover, consumer's and provider's policies are not needed because WS-Agreements suits for all the purposes they are required for. To evaluate QoS parameters during execution of composed Web Service and its reconfiguration in case of QoS parameters agreed values violation, the addition of monitoring stage can be involved. Classic Ontology-based approach and the approach emerged from combination of Ontology-based approach and SLA-awareness are presented in Fig. 6.

Utilization of SLA in proposed approach provides generalized way of storing QoS parameters for service providers. It helps performing WSC on the stage of workflow enactment as well as on the stage of workflow analysis. Another advantage compared to classic Ontology-based approach is enabling the run-time monitoring of the composite service.

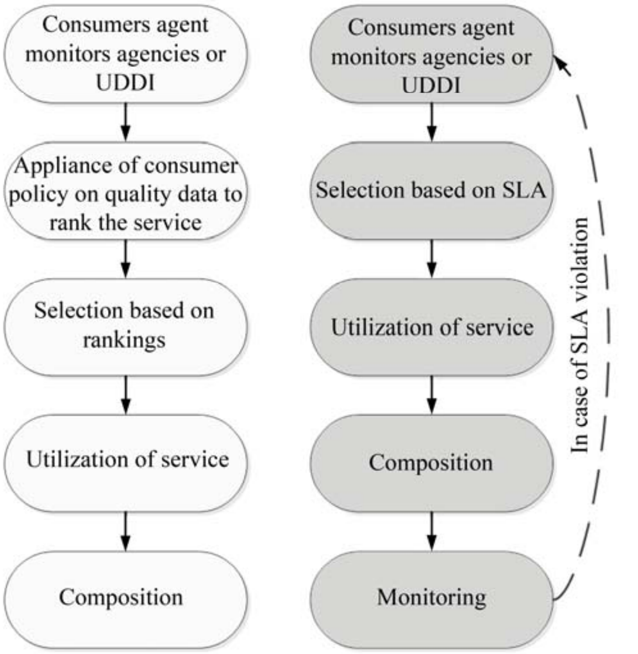

Fig. 6. Configuration of discussed approaches: (*a*) Classic Ontology-based approach; (*b*) SLA-aware WSC approach.

Service design tool

BPMA

The proposed workflow design method has been implemented based on the workflow design tool Business Process Modeling & Analysis (BPMA). It is realized using GTK+, the Dia diagram editor, PyDia interface, Python Interpreter, PyGTK (Set of Python wrappers for the GTK+ GUI library) and BPEA as a module for setting workflow parameters and analyzer. BPEA is a core BPMA component implementing the suggested workflow modeling and its analysis algorithms. BPMA functional scheme is shown in Fig. 7. BPEA includes five main submodules: “init”, “props”, “bplyzer”, “transform” and “reports”. Submodule “init” realizes Dia and user intercommunication. It provides the user interface, checks the user commands conformance and data correctness and also launches all module functions. The submodule “props” provides setting, changing and saving of workflows and their objects parameters. “Bplyzer” implements time minimization algorithms. The submodule “transform” implements the transformation logic to create a realization diagram from workflow requirements. The submodule “reports” generates and represents reports regarding workflow modeling and analysis results.

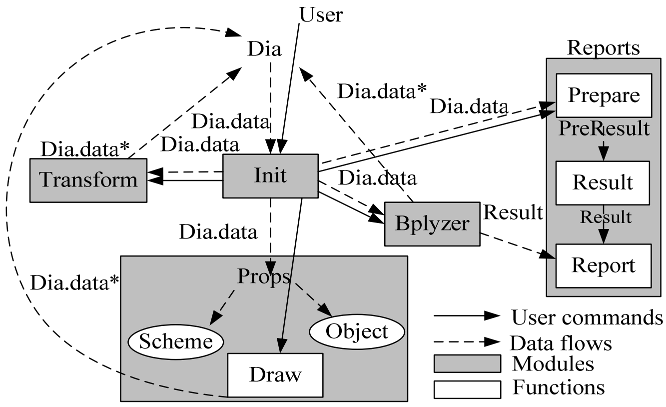

Fig. 7. BPMA functional scheme.

BPNE to BPEL transformation tool

The presented BPNE to BPEL transformation method is implemented in software tool with the structure shown in Fig. 8. The transformation tool performs: transformation of data extracted from BPNE model file into the representation form convenient for the subsequent work; translation of BPNE elements and components into BPEL code using its object-oriented model; gathering all translated elements and components in a single BPEL file.

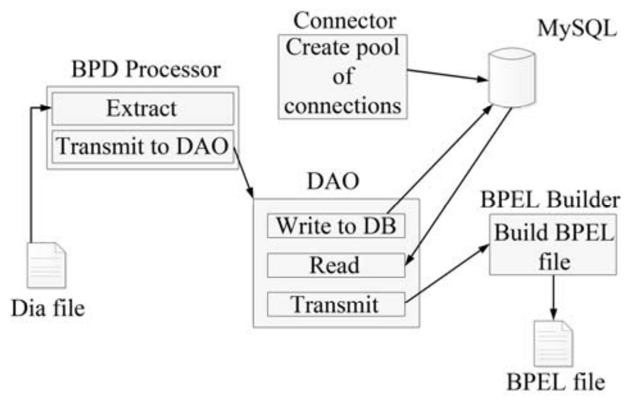

Fig. 8. Structural scheme of the BPNE to BPEL transformation tool.

Concept of Web-service composition module

Fig. 9 presents the structural scheme for software implementation of SLA-aware WSC on the stage of Workflow enactment. Software implementation for

SLA-aware WSC on the stage of Workflow execution: extracts the information about functional parameters from abstract BPEL-file; searches appropriate services in UDDI or service brokers; extracts SLA information through WS-Agreement; makes the decision which service to utilize based on WSA and Ontology Data; creates the list of QoS-aware services; performs composition into executable BPEL-file according to the information from the abstract BPEL-file.

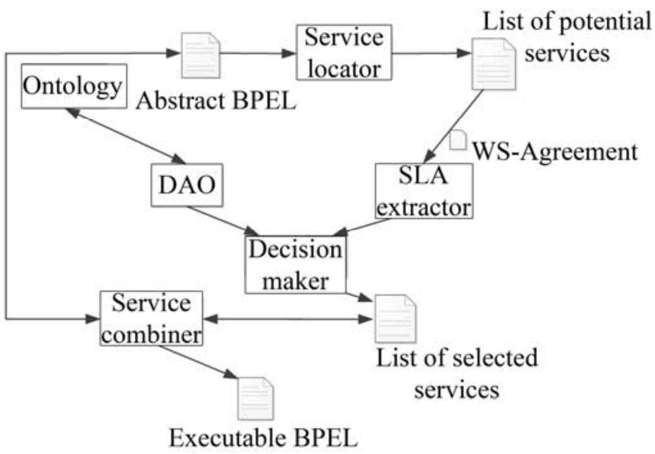

Fig. 9. Structural scheme for software implementation of SLA-aware WSC.

This scheme does not take into consideration the monitoring and dynamic reconfiguration of composite Web Service. Taking into account these factors, it should be modified or another software implementation specialized on Web Service monitoring can be used in order to provide its utilization.

Case-study

Workflow developed with the created tool is considered in [3]. BPMA has been tested during the planning and designing several services, such as “Bonus program”, “Selling Electronic Voucher”, “Tariff Plan Change” in the SITRONICS Telecom Solutions Company.

Tested results verify their ability to reduce time and costs of the service planning stage and service development in comparison with existing tools used for service design (BPWin, Enterprise Architect). Thus, for services having 1—1.5 months of development time, it can be reduced to 3—5 days and development costs can be decreased to 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.

Evaluation of the proposed NFR balancing method is demonstrated on charging of GPRS service extensively described in [15]. Table 5 shows FR decomposition of GPRS service.

Table 5. 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

GPRS service availability and delay were taken into account as two main NFRs which have direct influence on quality of provided services and operator’s key performance indicator. The specified values were obtained based on results of performance test campaign and specified in conventional units. The test campaign was performed by repeating common tests (about 5000 times) to get average values. Thus, considering statistical data and enterprise knowledgebase, GPRS service characteristics were estimated and the GPRS service target model shown in Fig. 10 was obtained using the proposed balancing method. For the delay, value 100 means 5 seconds as maximal response time for online request. For the availability, value 100 means 99.999% of service facility for real time systems.

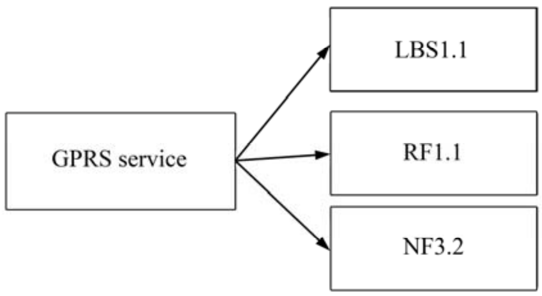

Fig. 10. Target model for GPRS service.

The target model provides the service optimal deployment configuration ensuring the service of the highest availability with a minimal delay. The proposed method is experimentally proved to be efficient for services design and deployment, providing NFR modeling, considering their influence on system functionality as well as getting service functionality with required functional and non-functional parameters.

Summary and outlook

A novel approach for analyzing and developing modern services provided with Web-oriented applications has been proposed. 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 system adaptation up to 10% by improving and automating the service planning stage. The proposed NFR balancing method provides description of NFR considering collaboration between services. The method increases efficiency of development process on testing and deployment stages and allows fast system reconfiguration on customer demand.

Modified BPMN to BPEL transformation method and tool providing consideration of workflow required parameters at abstract workflows transformation have been proposed. Based on analysis of current Web Service Composition approaches, it is shown that none of existing methods is able to meet all QoS characteristics. A novel SLA-aware method for Web Service Composition which satisfies all requirements has been proposed. The proposed approach allows performing dynamic composition based on SLA providing required values of QoS parameters, improving general QoS and decreasing service development and re-engineering time.

Future work will focus on research of monitoring stage in order to provide monitoring rules for all QoS parameters. Another important issue is to specify selection method for services based on SLA. Future work will focus on analysis of computing workflows and their enactment, services monitoring and re-configuration, which are strongly adjusted to the design and analysis stages. NFR balancing method will be extended with possibility to consider feasible change of NFR list and their priorities during different time period and also take into account changing priority between services.

Monitoring stage will be improved in order to provide monitoring rules for all QoS parameters. Another important issue is to specify selection method for services based on SLA.

References

1. Börger E. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. — Springer, 2011. — 14 p.

2. Business process management: Where business processes and web services meet / W. M. P. van der Aalst, B. Benatallah, F. Casati, F. Curbera, H. M. W. Verbeek // Data and Knowledge Engineering. — 2007. — Vol. 61, N. 1. — P. 1—5.

3. A novel approach to increase efficiency of OSS/BSS workflow planning and design / T. Kot, A. Reverchuk, L. Globa, A. Schill // Lecture Notes in Business Information Processing. — Springer, 2012. — Vol. 117. — P. 142—152.

4. 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/

5. Non-functional requirements in software engineering / L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos. — Boston: Kluwer Academic Publishers, 2000. — 472 p.

6. Social Modeling for Requirements Engineering / E. Yu, P. Giorgini, N. Maiden, J. Mylopoulos. — Cambridge, MA: MIT Press, 2011. — 752 p.

7. Van Lamsweerde A. Requirements engineering: From system goals to UML models to software specifications. — John Wiley and Sons, 2009. — 650 p.

8. Mendling J., Lassen K., Zdun U. On the transformation of control flow between block-oriented and graph-oriented process modeling languages // IJBPIM. Special Issue on Model-Driven Engineering of Executable Business Process Models. — 2008. — Vol. 3, N. 2, — P. 96—108.

9. Klingemann J., Wasch J., Aberer K. Deriving service models in cross-organizational workflows // Proceedings of the 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises. — March 1999. — P. 100—107.

10. Quality driven web services composition / L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, Q. Z. Sheng // Proceedings of the 12th International Conference on World Wide Web. — May 2003. — P. 411—421.

11. Constraint driven web service composition in METEOR-S / R. Aggarwal, K. Verma, J. Miller, W. Milnor // Proceedings of the 2004 IEEE International Conference on Services Computing. — September 2004. — P. 23—30.

12. QoS-aware middleware for web services composition / L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. T. Kalagnanam, H. Chang // IEEE Transactions on Software Engineering. — 2004. — Vol. 30. — P. 311—327.

13. Aiello M., Giorgini P. Applying the Tropos methodology for analysing web services requirements and reasoning about Qualities of Services. CEPIS Upgrade // The European Journal of the Informatics Professional. — 2004. — Vol. 5. — P. 20—26.

14. Maximilien E., Singh M. A framework and ontology for dynamic web services selection // IEEE Internet Computing. — 2008. — Vol. 8. — P. 84—93.

15. Method of non-functional requirements balancing when service development / T. Kot, A. Reverchuk, L. Globa, A. Schill // Journal of Theoretical and Applied Computer Science. — 2012. — Vol. 6, N. 3. — P. 50—57.

16. Kot T., Globa L., Schill A. Applying business process modeling method when telecommunication services development. // Proceedings of International Crimean Conference on Microwave and Telecommunication Technology, September 2011. — Sevastopol: Weber, 2011. — Vol. 1. — P. 457—458.

Received in final form October 29, 2012