

A Novel Approach to Increase Efficiency of OSS/BSS Workflow Planning and Design

Tetiana Kot¹, Andrey Reverchuk³, Larysa Globa¹, and Alexander Schill²

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

² Technische Universität Dresden, Fakultät Informatik, Deutschland

³ SITRONICS Telecom Solutions, Czech Republic a.s.

Abstract. Nowadays, communication technologies and the range of mobile operator services are changing extremely fast. This results in the need for constant adaptation and expansion of OSS/BSS¹ used in mobile operator networks. Currently, adaptation and expansion 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 workflow design method proposed in this paper fills this gap. It employs a well-defined workflow and analysis model for developing and adapting OSS/BSS. The applicability of this novel approach is confirmed by a prototypically implemented design software tool which has been tested in a telecommunication enterprise. The developed tool provides automation of service planning, computational independent workflow design and its transformation into its realization model. The reduction of development time and thus necessary financial input has been proven based on our real-world experiments.

Keywords: OSS/BSS, service planning, workflow design, Dia editor.

1 Introduction

“The technology sector in general and the telecommunication companies in particular operate in increasingly competitive environments. The companies that survive and excel are those offering the most compelling range of products and services. Since the underlying technologies tend to offer similar features and functionalities, the only differentiation is the services created from these technologies. The method used to create a great service is service design.

Even to the professionals within the telecommunication and service provider sectors (i.e. the companies that provide telecommunication/Internet services), it is often difficult to articulate the concept and purpose of service design. However, I have seen so many projects and service developments fail because there were no service designers in the project team.”[1]

¹ Operation Support System/Business Support System.

Telecommunication companies are notable for technological effectiveness and innovation, constantly renovating their technologies and services by adopting OSS/BSS - complex software systems, providing telecommunication companies functioning) [2]. OSS/BSS adaptation and expansion is realized via service planning, workflow design, realization and re-engineering according to the changes of requirements to services and OSS/BSS features. Additionally, mobile operators tend to minimize time of service provision, using constantly improving communication technologies and software applications. Thus, mobile operators require means and tools for fast workflow design and reengineering during OSS/BSS runtime and for providing services within a minimal time.

OSS/BSS adaptation and expansion mainly concerns the development of service provisioning applications and includes:

1. service planning, defining service provision time and resources;
2. computational independent workflow design, defining the order of tasks while service provisioning;
3. computing workflow design, enactment, monitoring and optimization.

Currently, the service planning stage is performed manually. This does not result in good solutions because a lot of factors have to be considered for finding the optimal solution when planning differentiated services [3]. Furthermore, in the current state-of-the-art computational independent workflow design, being performed, using existing notations and tools on 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, and also document and information flows, supporting service provision, and on the other hand provides poor connection to system functionality, which should realize these workflows.

This paper describes a novel computational independent workflow design method, focusing on the workflow model and its analysis, allowing to automate service planning stages and to reduce the time and costs for OSS/BSS adaptation in general.

The paper is structured as follows: Section 2 contains state of the art analysis of workflow design notations and tools. Furthermore, service implementation technologies are described. Section 3 introduces workflow design methods, focusing on two core aspects: A computational independent workflow model, describing service provision and considering service provisioning parameters, required for its analysis, and a workflow analysis method, providing service provisioning time minimization. Section 4 presents a prototypical realization of the design method and highlights evaluation results of the developed tool. 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.

2 State of the Art and Background

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 are presented in section 2.1, workflow analysis methods and systems in section 2.2. Parameters, which should be taken into account at the planning stage are presented in section 2.3.

2.1 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. Analysis has shown that in practice computational independent workflows are usually designed using graphical notations such as BPMN 2.0², UML AD³, USLD⁴ and tools such as CA ERwin Process Modeler⁵, Enterprise Architect⁶ and MS Visio⁷.

USDL is sufficiently generalized, still under development stage and doesn't fully meet all the requirements of OSS/BSS 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 [4]. The central 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. No execution time parameters are considered. All further existing workflow modeling notations have this core criticism in common.

Nevertheless, BPMN, providing the ability of computational independent to computing workflows transformation (BPEL⁸ diagrams), being widespread in industry, can be applied as a basic notation for OSS/BSS computational independent workflow design. Thus, it is intended to extend it adding the missing concepts.

2.2 Workflow Analysis

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

1. Design time analysis (simulation and verification). Monte Carlo simulations can be used as well as Petri Nets analysis as there exists transformation approaches for BPMN [5], UML AD [6], EPC [7], BPEL [8] to Petri Nets with their further analysis. USLD diagrams can be analyzed using service ontological analysis [9].
2. Runtime analysis (for instance, process mining, based on the execution logs) [10].

Software tools such as Pegasus, Cactus, ASKALON, GLUE, etc. [11] are used for these analysis fields. All mentioned and analyzed current possibilities for this task stage are very limited. There are no tools available to automate the service planning

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 <http://office.microsoft.com/en-us/visio/>

⁸ Business Process Execution Language.

stage considering service provision time and resources. As an improvement, a graph model can be used to verify workflow diagrams connectivity, as presented in [12]. By this, a service provisioning time minimization can be realized. The methods, usually used on this stage, such as shortest path problem for a graph, are not applicable, due to the fact that workflows, describing service provisioning, contain parallel tasks, and each of them can have a few implementation variants. Thus, some other mathematical approach should be applied for this problem.

Lacks of existing tools and methods of workflow design and analysis make their usage while service planning and OSS/BSS workflow design complicated. The central criticism is that the requirements analysis stage is applied mainly in a manual manner.

2.3 Service Planning

In differentiated service models, used today by mobile operators, QoS⁹ is realized by marking packets based in the customers’ service class. In response to these markings, routers and switches use various queuing strategies to tailor performance to requirements. According to this, service provision time and resource vary potentially heavily.

Service provision, depending on content and communication technologies, is characterized by parameters such as QoS, acceptable service delay, capacity, depending on radio-technology (GPRS/EDGE¹⁰; LTE¹¹), etc. Besides, separate tasks of the workflow can have a few variants of realization. For instance, subscriber requests can be transferred using technologies such as GSM¹², GPRS, CDMA¹³; tariffing can be realized using internal online raters, internal offline raters or external raters. Their values define service provision time and resources, which are to be defined on the planning stage.

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

$$R = \sum_{k=1}^K r_{ki}^n \tag{1}$$

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

Thus, when applying service planning, operators work with the following data:

- R – total amount of resources required for service provision (workflow execution);

⁹ Quality of Service.

¹⁰ General Packet Radio Service/Enhanced Data rates for GSM Evolution.

¹¹ 3GPP Long Term Evolution.

¹² Global System for Mobile Communications.

¹³ Code Division Multiple Access.

- r_{kl}^n - resource, required for implementing task l of stage k using realization variant n ;
- ξ_{kl}^2 - execution time of task l of stage k using realization variant n .

For all task implementation variants, time dependence on the resource $\xi(r)$ is not linear. This should be considered at the planning stage during the computational independent workflow design.

3 Workflow Design Method

According to [13], 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:
 - forming workflow graph and verifying its connectivity;
 - workflow execution time minimization;
 - transformation of workflow to realization diagram.

The suggested method modifies the MDA¹⁴ approach on the business logic level (fig.1.).

3.1 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 engineering approach is presented.

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

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

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, characterizing 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:

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

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

¹⁴ Model Driven Architecture.

Furthermore, separate work models can be represented in a formal manner as:

$$O = (O, I^O, P) \tag{4}$$

where O is the set of identification objects; I^O is the set of informational objects; P is the set of parameters, characterizing service provisioning concerning separate work realization.

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

$$R_O^n = (N_R, \xi_{kl}^n(r_{kl}), r_{kl}^n) \tag{5}$$

$P^O_7 = P_{ad}$ - set of additional work parameters.

In more details the presented model is described in [14]. It will allow to perform workflow analysis at a planning stage, applying graph theory and optimization algorithm, represented below.

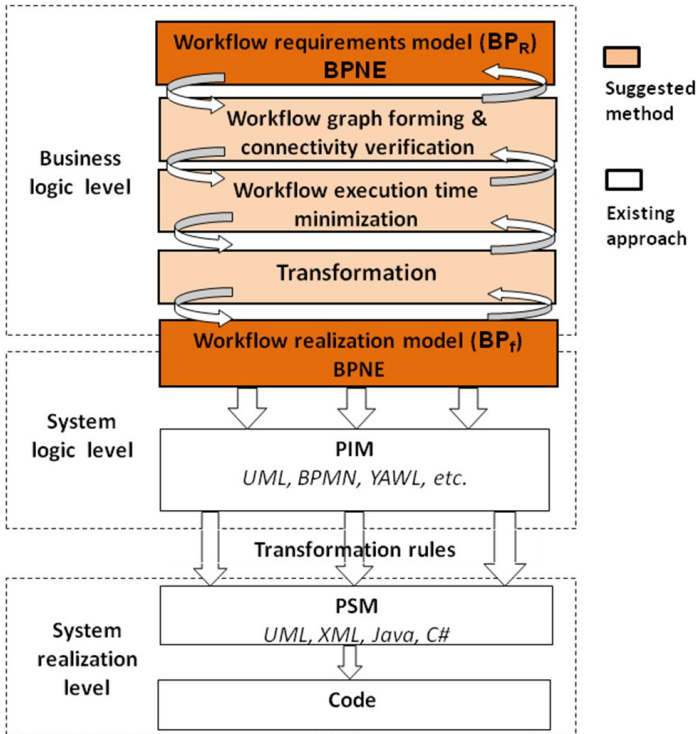


Fig. 1. Workflow design method extending the OMG MDA approach

3.2 Workflow Analysis Method

The workflow analysis method is the second central point of the proposed design method.

The workflow graph model can be represented as sequential stages, containing a few parallel executed tasks or just one task (fig. 2), which enables the definition of workflow execution time as follows:

$$T_{ex} = \sum f_{st} = \sum_k \max_l \xi_{kl}(r_{kl}); \tag{6}$$

The suggested method of workflow analysis, providing workflow model verification, execution time minimization and automating its transformation, can be represented mathematically as follows:

$$M = (G_f, G_v, M_{min}, M_{tr}) \tag{7}$$

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 representing the model to realization transformation. According to the proposed method, workflow analysis is performed as follows. When the workflow diagram is designed, having a few implementation variants of its operations, 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 limitation. After the time minimization procedure is realized, only one implementation variant for each operation is selected. The workflow model with minimal execution time is transformed into the realization model, describing the system modules, their functions, their time and resources requirements.

Generating the Workflow Graph and Its Connectivity Verification. The described workflow model allows requesting the control graph and information flow graph. Graph connectivity is verified using standard procedures and algorithms, described in.

Workflow Execution Time Minimization. Each task 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 execution time of the workflow (6), when the total resource (1) is limited and known.

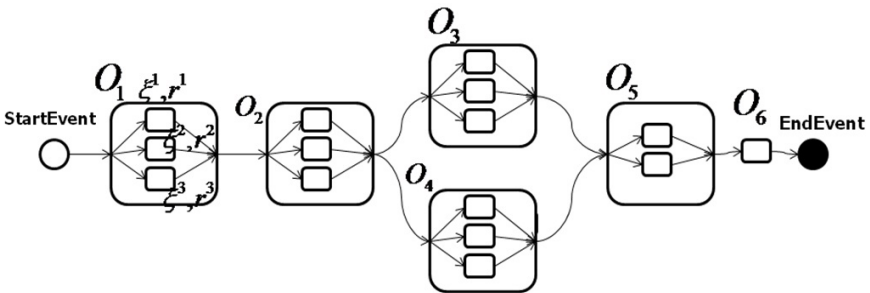


Fig. 2. Workflow graph with task implementation alternatives

According to practical experience, the number of typical software modules providing one service (implementing one workflow) is not more than 10. Thus, the number of parallel executed tasks can't exceed 10 and usually is not more than 5. Then the task of finding minimal workflow execution time is P-complete and can be solved in polynomial time.

The objective function of the task is represented in (8)

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

Furthermore, it is 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 is reached (8).

For finding the function specified in (8), dynamic programming is applied. This is realized in a two steps approach (fig.3):

1. determining minimal time of tasks executed in parallel works;
2. determining minimal time of workflow sequential stages, using results from the previous step.

Thus, using dynamic programming algorithms, the task of workflow analysis on the service planning and design stage, can be solved.

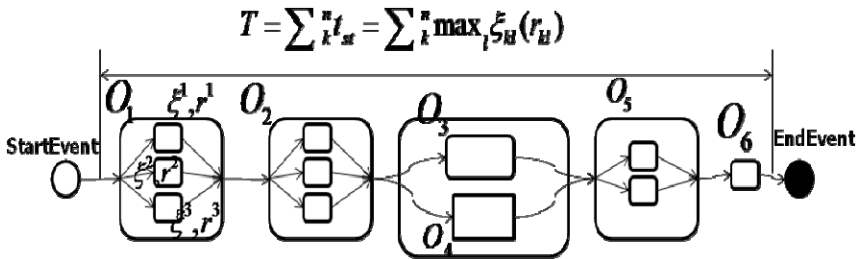


Fig. 3. Workflow time minimization

Transformation of Workflow Requirements into a Realization Diagram. The approach of transforming the workflow requirements into a realization diagram includes three core steps:

- 1) defining the ability of work to be automatically executed: $P^O_3 = a$ [yes/no/half];
- 2) generating the workflow intermediate model:
 - adding each of the task parameters – OSS/BSS subsystems and their functions;
 - analyzing task execution time and resources: dividing them in functional and non-functional components;

3) generating a workflow realization diagram, containing only functional components of time and resources, system modules and functions description.

The workflow realization diagram describes requirements to workflow realization and can be used for architecting the service provisioning application design.

4 BPMA

The proposed method has been implemented based on the workflow design tool BPMA¹⁵. It is realized using GTK+, the Dia diagram editor, PyDia interface, Python Interpreter, PyGTK¹⁶ and BPEA – a module for setting workflow parameters and analyzer. BPEA is a core BPMA component, implementing suggested workflow modeling and its analysis algorithms. Its functional scheme is represented in fig. 4. BPEA includes five main submodules: “init”, “props”, “bplyzer”, “transform” and “reports”. Submodule “init” realizes Dia and user intercommunication. It provides an user interface, checks user commands conformance and data correctness, and also launches all module functions. The submodule “props” provides setting, changing and saving of workflows and its 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.

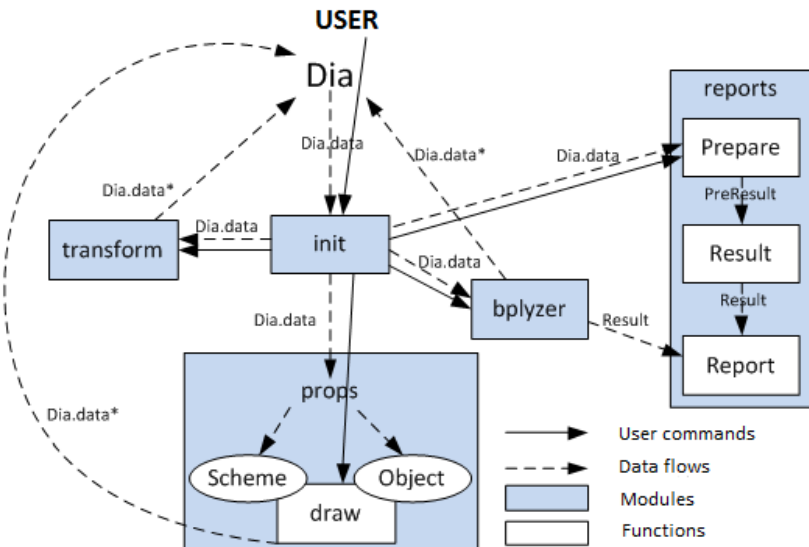


Fig. 4. BPMA functioning scheme

¹⁵ Business Process Modeling & Analysis.

¹⁶ Set of Python wrappers for the GTK+ GUI library.

Figure 5 visualizes a workflow developed with the created tool.

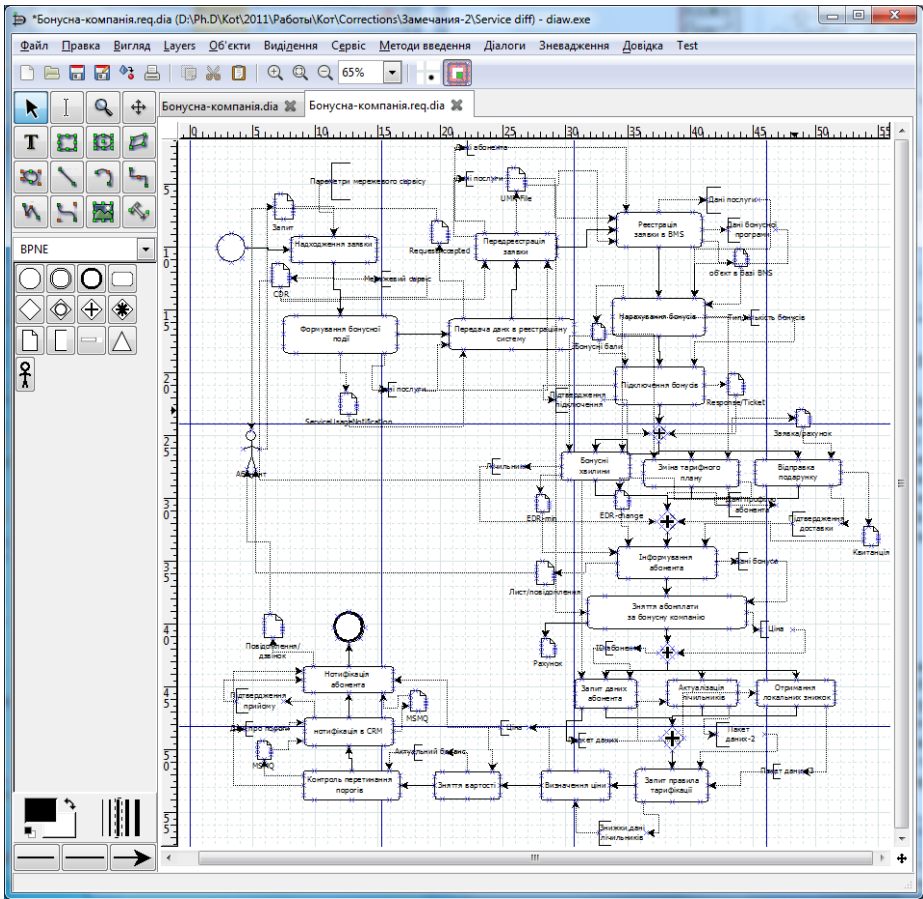


Fig. 5. Example workflow used within a case study

BPMA has been tested intensively during the planning and design of several services in the SITRONICS Telecom Solutions company. Testing results have proven its ability to reduce time and costs of the service planning stage and service development in general in comparison to existing tools, used when service design (BPWin, Enterprise Architect, etc.). Thus, 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.

5 Summary and Outlook

A novel approach for analyzing and developing OSS/BSS workflows has been proposed in this paper. The discussion focused on the development of a computational

independent workflow model and the procedure of its analysis. The proposed 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 OSS/BSS workflow design method reduces the time of OSS/BSS adaptation up to 10% by improving and automating the service planning stage.

Future work will focus on transforming the proposed workflow model to an executable one, providing required parameters transfer, its enactment, monitoring and optimization both on the design and on runtime stages.

References

1. Pang, S.: Successful service design for telecommunications: a comprehensive guide to design and implementation, p. 351. John Wiley & Sons Ltd. (2009)
2. Terplan, K.: OSS Essentials: Support System Solutions for Service Providers / Kornel Terplan, p. 610. John Wiley, New York (2001)
3. RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, <http://tools.ietf.org/html/rfc2474>
4. Börger, E.: Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL, p. 14. Springer (September 2011)
5. Transformation of BPMN models for Behaviour Analysis, http://www.win.tue.nl/~jmw/_media/public/transformationforbehaviouranalysis.pdf
6. Baresi, L., Pezzè, M.: On Formalizing UML with High-Level Petri Nets. In: Agha, G., De Cindio, F., Rozenberg, G. (eds.) APN 2001. LNCS, vol. 2001, pp. 276–304. Springer, Heidelberg (2001)
7. Nüttgens, M., Feld, T., Zimmermann, V.: Business Process Modeling with EPC and UML: Transformation or Integration? In: Schader, M., Korthaus, A. (Hrsg.) Proceedings of the Unified Modeling Language - Technical Aspects and Applications, Mannheim, Heidelberg, October 1997, pp. S250–S261 (1998)
8. Ouyang, C., Verbeek, E., van der Aalst, W.M.P., Breutel, S., Dumas, M., ter Hofstede, A.H.: Formal Semantics and Analysis of Control Flow in WS-BPEL. Technical report (revised version), Queensland University of Technology (October 2005)
9. Oberle, D., Bhatti, N., Brockmans, S., Niemann, M., Janiesch, C.: Countering Service Information Challenges in the Internet of Services. *Journal of Business & Information System Engineering* 1(5), 370–390 (2009)
10. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: Workflows for e-Science, Scientific Workflows for Grids, 1st edn., p. 552. Springer (2006)
11. van der Aalst, W.: BPM and Workflow Analysis. *BPTrends* 5(4), 1–2 (2007)
12. Globa, L., Kot, T., Schill, A., Strunk, A.: Method of IBIS design and workflow realization. *Polish J. of Environ. Stud.* 18(4a), 35–38 (2009)
13. Ko, R.K.L., Lee, S.S.G., Lee, E.W.: Business Process Management (BPM) Standards: A Survey. *Business Process Management Journal* 15(5), 48 (2009)
14. Kot, T., Globa, L., Schill, A.: Applying business process modeling method when telecommunication services development. In: Proceedings of 21st International Crimean Conference Microwave and Telecommunication Technology, CriMiCo 2011, Sevastopol, Ukraine, vol. 1, pp. 457–458 (2011)