

THE SELECTION OF THE ARCHITECTURE OF ELECTRONIC SERVICE CONSIDERING THE PROCESS FLOW

PETERIS STIPRAVIETIS, MARIS ZIEMA

*Institute of Computer Control, Automation and Computer Engineering, Faculty of Computer Science and Information Technology, Riga Technical University, Latvia
{Peteris.Stipravietis, Maris.Ziema}@rtu.lv*

Abstract: *The article discusses the basic steps of the electronic service design method using several languages, transformations between them and simulation. The inclusion of method which segments the process activity graph in these steps is evaluated. Merge requirements taking into account the process control flow are proposed to be implemented in the segmentation method so that could be used to solve problems concerning the selection of electronic service business process architecture. The solutions provided by this method can be interpreted as BPEL orchestrations, defining which activities could be implemented as web service calls. Using the proposed approach requirements can be defined in input data of segmentation method as well as during the execution of it. Possible solution of the implementation of requirements is also provided.*

Keywords: Quality driven attributes, Web services, Orchestration.

Introduction

E-services are common in information society nowadays, and even though they tend to become more and more accessible and varied, the problems that occur during the design phase of the service remain the same. These problems include, for example, questions on how to facilitate the creation of business process to the user with no specific programming skills, how to define the process in a way that creates the process description abstract but accurate enough at the same time, how to check the created model – to determine the weaknesses, perform the measurements based tuning, and others – for instance, designing the architecture of the process so that it conforms to various requirements. The design of architecture of the process may be influenced by multiple criteria – the requirements of performance, costs of development and maintenance, the reusability possibilities of the process or its parts and others. During the design phase of the process one should use an approach that allows the analysis of the process model designed and provides appropriate changes of the process structure, instead of modelling the process while trying to anticipate all possible requirements and restrictions – it would be easy enough with one criterion to comply with and almost impossible in case of five concurrent and conflicting criteria. Hence the prerequisite – the approach or methodology used during the design of the business process should allow to define various requirements and change the impact of these requirements on the process, thus providing effective way to model different process structures which implement the same functionality, but the possibility of the analysis and simulation of the business process model relies on the choice of the language used to describe the process.

Proposed approach

Existing business process modelling languages can be divided in two groups. The languages of the first group are favoured by the academic community, but rarely used in real-life solutions. These languages are based on Petri nets, process algebra; they have formal semantics, which allow the validation of the models described by these languages. The languages of the second group are used in real-life projects much more than in academic researches. BPEL, WSFL and WSCI are among these languages. These, so called business languages, often lack proper semantics, which could lead to debate on how to interpret the business models described by these languages. The availability of different implementations of these languages from different vendors does not facilitate the situation either, yet they are used much more, compared to seldom used models described by academic languages. If a situation arises when business process model described by business language needs to be validated using Petri nets, one must either abandon the validation or transform the process model to another model, described in academic language, for example YAWL. The authors propose reverse approach – first, a process is created using academic language. The design problems of the process model can then be solved by mathematical means. Second, the verified and updated model is transformed to model described in business language. The advantages of the approach described follows:

- If a model is created using academic language, it is more readable and maintainable than the model, which is a transformation result itself. It is also easier to perform analysis of untransformed model, because the transformation could lose some design information.
- Model, transformed to business language, is already validated and ready to be executed. Of course, the model must be double-checked to make sure if it needs any corrections. The alternatives of the

execution environment for the model are much more than the environments for academic languages; in addition to that, they have superior technical support.

The model of the business process can be changed and improved during the transition from one language to another – the approach proposed by authors consists of following phases (Fig. 1, dotted lines show that some phases may be omitted):

- The design of the business process using academic language;
- The validation and simulation of the business process model designed;
- Transformation of possibly improved model to primitive structure;
- The segmentation of primitive structure represented by graph using Quality Attributes Driven Web Services Design (QAD WS) method, developed by Zeiris (2009);
- The transformation of primitive structure (or its segmentation) to process using business language.

The purpose of this article is to examine the 4th phase of proposed approach – the usability of QAD WS method in this phase to generate multiple process structures corresponding to quality attributes given, and identify possible improvements of the method to ensure of its usefulness and efficiency during this phase.

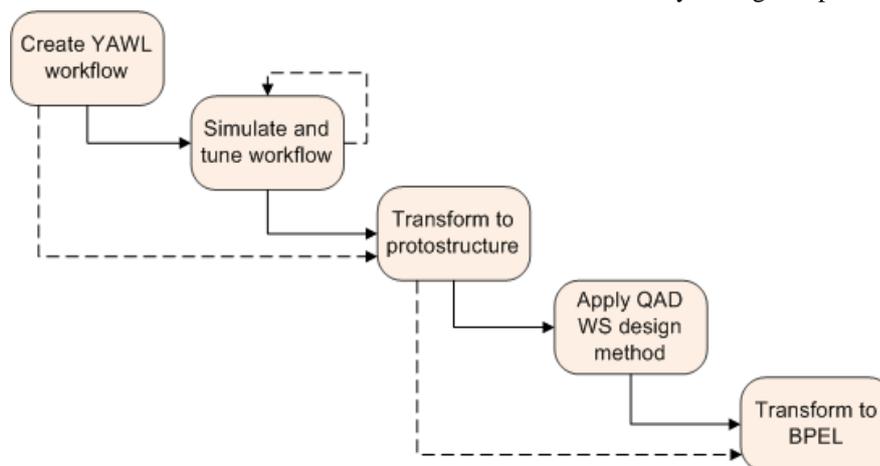


Fig. 1. The phases of proposed design approach

Similar approach is proposed by Pornudomthap and Vatanawood (2011): their solution is based on straightforward conversion of YAWL workflow to BPEL process. The approach proposed in this article allows transforming the workflow to any business process language and to simulate and optimize the process flow as well.

The first step is the design of the business process using academic language. The initial business process model is created during this phase. Designer should use only constructions supported by language chosen as a business language, i.e., avoid ‘goto’ workflow pattern if a business language selected is BPEL which does not support arbitrary loops. The workflow patterns supported by major business process languages are discussed by Havey (2005).

The second step consists of the simulation and tuning of business process model. Rozinat et al (2008) propose simulation which uses process design data, historical data about executed process instances from audit logs and state data of the running process instances from the execution environment. Data from all three sources are combined to create simulation model – design data are used to define the structure of the simulation model, historical data define simulation parameters, state data are used to initialize the simulation model.

Altering the simulation model allows to simulate different situations, for example, to omit certain activities or divert the process flow to other execution channels. Taking into account the state data of running process instances, it is possible to render the state of the system in near future and use the information to make decisions regarding the underlying business process.

The simulation of the workflow is carried out using process data mining framework ProM, proposed by van der Aalst et al. (2007). To create simulation model, following steps are performed:

- Workflow design, organizational and audit log data are imported from execution environment;
- According to imported data a new YAWL workflow model is created and state data are added;
- The new model is converted to Petri net;
- Resulting Petri net is exported to simulation execution environment CPN Tools as a colored Petri net – an approach proposed by Jensen et al (2007).

Stipravietis and Ziema (2011) have discussed the suitability of this simulation method to the process design approach in their work.

The third step provides the transformation to primitive structure. Primitive structure is simplified definition of business process control flow, although it can also be used to maintain the data flow. The primitive structure

serves as an intermediate between academic and business languages and can be used to create processes described by multiple languages, not only BPEL. The primitive structure may be changed and improved during this phase to facilitate the transition to target language, i.e., restructure its control flow in a way that it becomes well-formed and contains only patterns supported by BPEL. The creation of process primitive structure and its properties are discussed by Stipravietis and Ziema (2010).

The fourth step provides the segmentation of primitive structure using the Quality Attributes Driven Web Services Design (QAD WS) method which offers the segmentation of business process, represented as oriented graph. The segmentation result depends on process quality attributes selected by designer and their respective values. The result of this method is Pareto optimality set – the method returns the most suitable segmentations from all possible considering the quality attributes given.

The business language selected by authors and used in their proposed approach is BPEL, and using of QAD WS method on primitive structure would provide the possible structures of BPEL process – which parts of the process would belong to orchestration and which ones would be implemented as web service calls. Rosario et al (2006) also discuss the partitioning of Web services into orchestrations based on their QoS values, but their approach do not use multicriterial optimization – that approach is based on Petri nets and usage of statistical data.

The last phase of proposed approach is the transformation of primitive structure to business language process which results in the business process defined in business language. This process is not ready to be executed, but its structure corresponds to initial process model described by academic language and maintains its process flow. The transition from primitive structure to BPEL is proposed by Stipravietis and Ziema (2010).

The overview of QAD WS method

The QAD WS method perceives the business process as an oriented graph G , whose vertices corresponds to process activities, but edges between them represents the control flow. Using various quality attributes and the structure of graph G , QAD WS method solves multi-criteria optimization task, which results in the segmentation set of initial graph G : $G' = QAD(G)$. According to Zeiris and Ziema (2007), criteria used by the method are:

- Costs of development C ;
- Performance T ;
- Maintenance costs E ;
- Reusability R ;
- Integrity I .

The segmentation set G' consists of N most optimal solutions designer can choose from – in principle this method greatly reduces possible solutions of process architecture, thus aiding the designer.

The result of the 3rd phase of authors proposed approach is primitive structure – oriented graph P that corresponds to initial YAWL workflow, which could be used as an input graph G for QAD WS. Authors note that P is more complicated as G – in addition to process activities and links between them it also contains the process control flow.

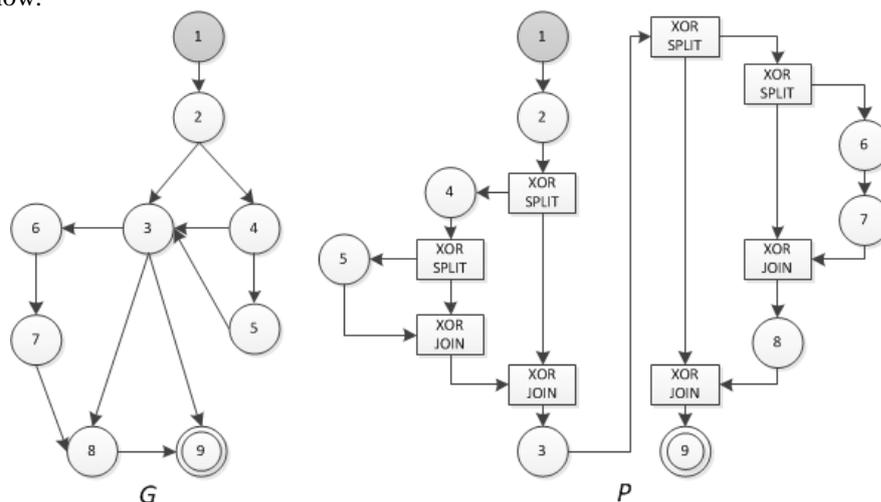


Fig. 2. Example – graphs G and P

Fig. 2 shows both graph G usable as QAD WS method input, and corresponding primitive structure graph P . Structure of both graphs are virtually identical, although P also contains control flow elements, in this case exclusive choices (XOR). Note that G does not define split types and therefore does not contain control flow information. Taking this into account, authors conclude that current implementation of QAD WS method provides results which does not preserve control flow logic of initial process.

This conclusion is confirmed by practical tests of QAD WS method – if one must find the segments of P taking into account only reusability, QAD WS method returns one solution, where all vertices of P form their own segment. In other words, QAD WS method propose that all activities must be implemented as web service calls, ignoring the fact that some activities does not perform any work but serve only as control flow providers. This solution is not only wrong but also illogical – how would one design a parallel flow in BPEL, if both start and end of it should be separate web service calls? The answer in this case should be either that entire parallel flow block is implemented as single web service call or that activities of each parallel branch are segmented at choice but the start and end belong to orchestration. The QAD WS method supports the concept of restriction, although current version provides only exclusive restrictions, i.e., activity A cannot be segmented with activity B. Hence the conclusion – in order to be properly used on selecting the process structure, the QAD WS method must support inclusive restrictions guaranteeing that start and end activities of some block N of graph P belong to one segment.

The improvements of QAD WS

Current implementation of QAD WS method processes both input and output data as graphs defined in XMI (XML Metadata Interchange). Restrictions (which vertices cannot appear in one segment) are defined during the method runtime. This approach allows changing dynamically the method parameters but cannot be used to define merging restrictions, and manual merge of control flow vertices can lead to erroneous results.

Authors propose to extend the restriction concept used in QAD WS. The restrictions are indices, describing segments – QAD WS should be able to merge vertices which have the same index. Let us define two types of restrictions – initialization and runtime restrictions. Initialization restrictions are defined in input data and are read-only during the method execution, but runtime restrictions are defined during the method execution and can be altered at will. Each type of restriction sets consists of merge restrictions and merge prohibitions:

$$I(G) = U_I(G) \cup J_I(G), W(G) = U_W(G) \cup J_W(G);$$

$I(G)$ – initialization restrictions, $W(G)$ – runtime restrictions, $U_x(G)$ – merge prohibitions, $J_x(G)$ – merge restrictions.

Both merge restrictions and prohibitions are mutually exclusive:

$$U_x(G) \cap J_x(G) = \emptyset$$

Initialization restrictions present in input graph G must be present in each segmented graph Γ :

$$\text{If } I(G) = U_I(G) \cup J_I(G), \text{ then } I(\Gamma) = U_I(\Gamma) \cup J_I(\Gamma) = I(G) \quad \forall \Gamma: \Gamma \in QAD(G)$$

Runtime restrictions cannot contradict with initialization restrictions, i.e., following condition must be true:

$$U_W(G) \cap J_I(G) = \emptyset \wedge J_W(G) \cap U_I(G) = \emptyset$$

If this condition is not true, QAD WS method must return empty result set.

Due to new restrictions, the upgraded QAD WS method must not yield results which contain overlapping segments.

Current implementation of QAD WS defines that no segment from result set has common vertices with other segments, i.e., if graph $\Gamma \in QAD(G)$ consists from segments S_1, S_2, \dots, S_n , then

$$S_i \cap S_j = \emptyset: S_i \in \Gamma, S_j \in \Gamma, i \neq j.$$

After the introduction of merge restrictions, the segments may contain one another but overlapping must still be avoided:

$$S_i \cap S_j = \emptyset \vee S_i \cap S_j = S_i \vee S_i \cap S_j = S_j : S_i \in \Gamma, S_j \in \Gamma, i \neq j$$

After defining the requirements for QAD WS improvements, authors propose a way to define them in input data. As mentioned before, QAD WS method receives, processes and returns data in XMI format. To define restrictions in XMI representation of graph, underlying XMI schema must be changed. A restriction is simply an index and is used as an attribute of vertex – to be able to define initialization restrictions, XMI schema fragment describing the vertex must contain optional attribute “Segment”. If this attribute is omitted, vertex can be merged in every possible segment (avoiding overlapping, of course). If this attribute has value, then all vertices with the same index value must be merged in the same segment.

The example below is a fragment from XMI formatted graph – it describes four vertices and their merge restrictions using the attribute “Segment”: Activity_1 and Activity_4 belongs to segment “1”, Activity_2 belongs to segment “2”, but Activity_3 has no restrictions – it can be merged both in segment “1” and segment “2”, as well as form its own segment.

```

<subvertex xmi:type="uml:State" xmi:id="Ub85d2fba-82dc-4761-9d8e - 9449884adfa6" xmi:uuid="b85d2fba-82dc-4761-9d8e - 9449884adfa6" name="Activity_1" Segment="1"/>
<subvertex xmi:type="uml:State" xmi:id="U4122a5e7-c725-4aa5-8486-e1ae5cc5c77b" xmi:uuid="4122a5e7-c725-4aa5-8486-e1ae5cc5c77b" name="Activity_2" Segment="2"/>
<subvertex xmi:type="uml:State" xmi:id="U8d7cf0ad-dc8a-4ddd-983b-c88b6aea44a4" xmi:uuid="8d7cf0ad-dc8a-4ddd-983b-c88b6aea44a4" name="Activity_3"/>
<subvertex xmi:type="uml:State" xmi:id="U5d900b52-cf99-4a84-bd5c-b66c927b2e9a" xmi:uuid="5d900b52-cf99-4a84-bd5c-b66c927b2e9a" name="Activity_4" Segment="1"/>

```

Presuming that activities are executed sequentially, QAD WS method offers three solutions of the example which conform to restriction requirements (Fig. 3).

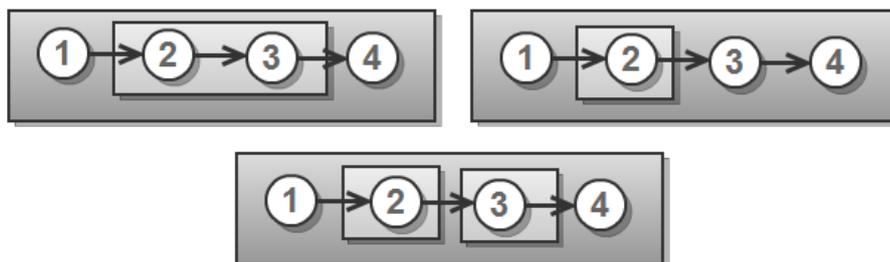


Fig. 3. QAD WS solutions of given example

Summary

During the design of electronic service systems one must take notice of various quality requirements of the implementation of the business process, for example, performance, reusability and others. The Quality Attributes Driven Web Services Design (QAD WS) method allows defining the impact of the said requirements. This method solves multi-criteria optimization task and offers Pareto optimality set. The authors have chosen this method as a part of their electronic services system design method and use it to prepare a business model for the transformation to the business language, in this case BPEL. The usage of BPEL lets the results of the QAD WS method be interpreted as different process orchestrations. Unfortunately the QAD WS method lacks the ability to preserve process control flow – this can lead to incorrect solutions.

Taking into account these shortcomings and improvements proposed, the authors set the tasks to improve the QAD WS method:

- Define XMI schema which allows to describe merge restrictions of a vertex;
- Implement the changes of the QAD WS method – it should be able to process and interpret the given initialization restrictions, i.e., from the process graph G and its initialization restriction set $I(G)$ it must produce the segmented graph $G' = T(G, I(G))$;
- During the processing of G , following must be taken into account:
 - Runtime restrictions cannot contradict with initialization restrictions;
 - Partial overlapping of the segments is not acceptable;
 - The initialization restrictions must be preserved and be present in every resulting graph.

To select appropriate solution from the QAD WS Pareto optimality set, usually the non-automatic analysis methods are applied. Using various Web service selection approaches – the WS selection depending on data types used, as shown by Stipravietis and Ziema (2007) and the description of the process they implement, proposed by Stipravietis and Ziema (2008)– it is possible to further narrow down the solutions which must be processed manually.

References

- Havey, M., 2005. „Essential Business Process Modeling” O’Reilly 141 p.
- Jensen, K., Kristensen, L.M. and Wells, L., 2007. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. International Journal on Software Tools for Technology Transfer, vol. 9(3-4), pp 213-254
- Pornudomthap, S. and Vatanawood, W., 2011. Transforming YAWL workflow to BPEL skeleton. Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference, Beijing, China, pp 434-437
- Rosario, S., Benveniste, A., Haar, S. and Jard, C., 2006. Foundations for Web Services Orchestrations: Functional and QoS Aspects, Jointly. Leveraging Applications of Formal Methods, Verification and Validation. ISoLA 2006, Second International Symposium, pp. 309-316

- Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M. and Fidge, C. J., 2008. Workflow Simulation for Operational Decision Support Using Design, Historic and State Information. Proceedings of the 6th International Conference on Business Process Management (BPM 2008), 2008, Milan, Italy. LNCS, vol. 5240, Springer, pp 196–211
- Stipravietis, P. and Zieme, M., 2007. Identification of Web Services Using Extended Metadata. Scientific Journal of RTU, 5th series, Datorzinātne, vol. 32, pp 108-115
- Stipravietis, P. and Zieme, M., 2008. Selection Of Web Services Using Industry Standards UML And XMI. Scientific Journal of RTU, 5th series, Datorzinātne, vol. 35, p. 118-125
- Stipravietis, P. and Zieme, M., 2010. The Design of Electronic Service Process Using YAWL. Proceedings of the Ninth International Baltic Conference, DB&IS 2010, Univesity of Latvia Press, pp 75-90
- Stipravietis, P. and Zieme, M., 2011., M. The Design of Electronic Service Process Using YAWL and CPN Tools. Selected Papers from the Ninth International Baltic Conference, DB&IS 2010, IOSPress, pp 183-196
- Van der Aalst, W.M.P., van Dongen, B.F., Gunther, C.W., Mans, R.S., Alves de Medeiros, A.K., Rozinat, A., Rubin, V., Song, M., Verbeek, H.M.W. and Weijters, A.J.M.M., 2007. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn and A. Yakovlev, editors, Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007), LNCS, vol. 4546, Springer, pp 484-494
- Zeiris E. and Zieme M., 2006. Criteria for Architecture Estimation and Architecture Selection of E-Services System. Scientific Journal of RTU, 5th series, Datorzinātne, vol. 27, pp 91-98
- Zeiris, E., 2009. E-service system architecture, Ph.D. thesis, Riga Technical University, Riga, Latvia