Tool for Interactive and Automatic Composition of Business Processes

Aleksandra Koleva[1] and Aleksandar Dimov[2]

Abstract – **Web services provide means for electronic execution of business processes. However, composition of services for this purpose may become a tedious task for non-IT professionals. In this paper we propose an approach to cope with this problem by developing a web-based software tool that is capable to propose to users appropriate services for creation of a business process. Afterwards the tool automatically compose and executes the selected services.**

Keywords – **Service Oriented Architecture, Service Discovery, Service Selection, Automatic Service Composition, Business Process.**

I. Introduction

Service Oriented Architecture is an architectural concept for development of distributed systems that has matured a lot during the last two decades [4]. It is claimed to enable reusability and straightforward integration of highly complex heterogeneous systems. The building elements and fundamental notion of SOA systems are called services. As stated in [1] service is a broad term that has different meaning depending on its usage context. For example, in computer science, the terms of web-service, e-service and business service have common meaning. In this paper, we will focus on web-services (WS), which are broadly recognized as [12]: Loosely coupled reusable software components that encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet and XML-based protocols. Web services should be considered as software components, accessible by unique URL. Interfaces of such components are described using well-defined protocols, majority of which are XML-based. Web-services are also composable and may be consumed by other services for use in different applications or in execution of business processes [5].

However, currently integration of services is mainly performed by technically aware personnel and general users doesn't have enough possibilities to create and execute business processes of their own. This paper presents an interactive tool that is capable to both provide recommendations to the users for service composition and also automatically implement and execute it. For this purpose it provides users with recommendations about services to be included into the composition, this way facilitating the process of their selection. This is based on information about service functionality. Afterwards, it automatically binds the selected services, asks for input data, if needed and finally, executes them.

The rest of the paper is organized as follows: Section 2 briefly presents current state of the art in the area; Section 3 describes some design and implementation details about the tool for interactive automatic composition of business processes and finally Section 4 concludes the paper and points out directions for future research.

II. State of the Art in Service Composition Discovery and Selection

There exist several research directions that have relation with the work presented here. Among them are service composition, discovery and selection. Two big groups of approaches for service discovery and selection may be distinguished – based on semantic information and the second, based on information about Quality of Service (QoS). It should be noted that concerns finding appropriate service, based on its functionality. On the other hand, QoS approaches, mostly focus on selection among services, when you have several alternatives that fulfill functional requirements. Approaches that use semantic information usually involves tools for automated web service discovery based on some description of the service semantics using a language like OWL-S [6] or analysis of the service WSDL [7]. Service selection, based on QoS use a formal representation of service quality and corresponding models to select the best service that fit user needs [9] or a given composition [10].

As pointed out in the introduction, in order to achieve substantial business value from services it is needed to be able to compose them into bigger units. There exist several classifications [2], [3], and [11] of service composition approaches, which discuss the subject in deep details. However for the purpose of this work we focus on more fundamental service composition classes, which are briefly described below:

Dynamic and static composition – dynamic composition may also be referred as run-time or automatic composition, where respective services are composed automatically during execution of the composite service. In order to achieve this, a number of additional disciplines as artificial intelligence, formal specification and etc. may be involved. Dynamic composition allows for SOA systems to change themselves transparently for the users and adapt to change in the environment and/or requirements.

[1] Aleksandra Koleva is with the Faculty of Mathematics and Informatics at Sofia University "St. Kliment Ohridski", 5 James Bourchier Blvd, Sofia 1164, Bulgaria, E-mail: alexandra.koleva90@gmail.com

[2] Alexandar Dimov is with the Faculty of Mathematics and Informatics at Sofia University "St. Kliment Ohridski", 5 James Bourchier Blvd, Sofia 1164, Bulgaria, and Section of Software Technologies and Information Systems, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Acad. Georgi Bonchev Str., Block 8, 1113 Sofia, E-mail: aldi@fmi.uni-sofia.bg

Different approaches for dynamic service composition are used. For example, model languages, like UML may be used for high level abstract description of the composition. In the approach, described in [8], Object Constraint Language (OCL) is applied to enable transformation to particular service composition standards like WS-BPEL or OWL-S. Another aspect of dynamic composition is about the method for selection of services that will take part in the composition. It may be based on Quality of Service (QoS), on semantic discovery of services or eventually, on combination of both approaches.

On the other hand static composition deals with design time interconnection of services. It requires advanced users and/or a certain amount of programming efforts in order to be implemented.

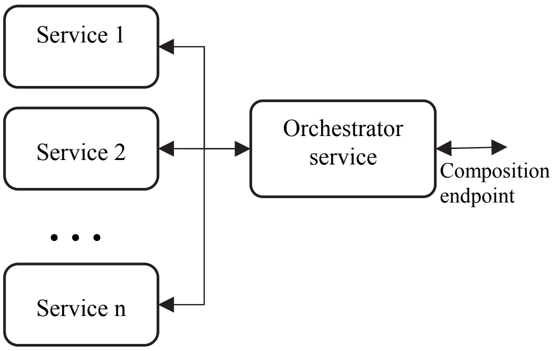

Fig. 1. Service orchestration

Choreography and orchestration – these two composition approaches focus more on behavioral details about role of specific service in the composition. Service orchestration is about organization of a centralized service, which executes as the main business process. It is called orchestrator and acts as an endpoint for the other services that take part into the composition. It also coordinates their execution (Fig. 1).

Service choreography (Fig. 2) is more about distributed management of invocation of services that participate in the composition. This is in contrast with orchestration, where control flow from orchestrator's perspective is important. Choreography is composition of services, where interactions between multiple services are of prime importance.

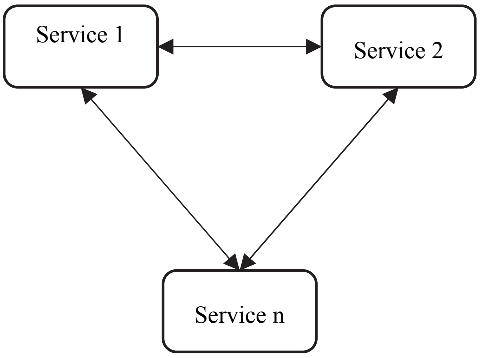

Fig. 2. Service choreography

Service composition notations – There exist a number of notations to describe service compositions as well as other languages that supplement them. The most commonly used service composition language is called WS-BPEL (Web Services Business Process Execution Language) [13]. It is aimed towards description of service compositions based on XML and WSDL variables, and structured composition representations using standard operators as logical branches, loops and etc. In order to achieve automatic compositions WS-BPEL should be complemented with a corresponding notation, which is machine readable and would enable programmable search and composition of services. One such notation is OWL-S [14], which represents a technology to provide semantic markup for web services. It is aimed towards description of the semantics of services and is based on three main sub-ontologies:

- Profile – describes what actually the service will do
- Process model – describes how the service works and helps for implementation of processes for service invocation, composition and monitoring.
- Grounding – describes how the service can be accessed.

There also exist other semantic notations for services, like WSMO (Web Service Modeling Ontology) [15] and SWSF (Semantic Web Services Framework) [16].

So far in this section, we have reviewed some of general properties of existing approaches for service discovery and composition. Further we will make an overview of the work directly related to ours – these are the existing service composition platforms.

Cloudwork [17] is an integration platform that allows customers to automate tasks between existing web- and cloud-based systems. For example, one cloud automate businesses with ticket support system or synchronize two storage services (like Google Drive with Dropbox).

Elastic.io is an API integration and orchestration platform, which offers simple interface to create and execute multiple web applications directly via the browser. It is appropriate to be used by nonprogrammers, however does not allow free access.

Although not directly aimed at service composition, Wufoo [18] helps users to build online forms, share them and collect data. It is Internet application which automatically builds the database, backend and scripts needed to make collecting and understanding user‘s data easy.

"If This Then That" [19] gives users possibility to connect applications via models and triggers, expressed in the so-called "recipes". They may be either of type "if recipe", allowing user to create connection with statement "if this then that", or "do recipe" enabling simple actions over the web. Users have to activate accounts for the relative connected channels like e-mail address, Dropbox account, google drive etc.

Zapier [20] uses a combination of triggers and activities (or action-reaction rules), called zaps to allow users to define combinations of applications. A drag and drop environment for creation of zaps is provided and the latter may be also scheduled for iterative execution.

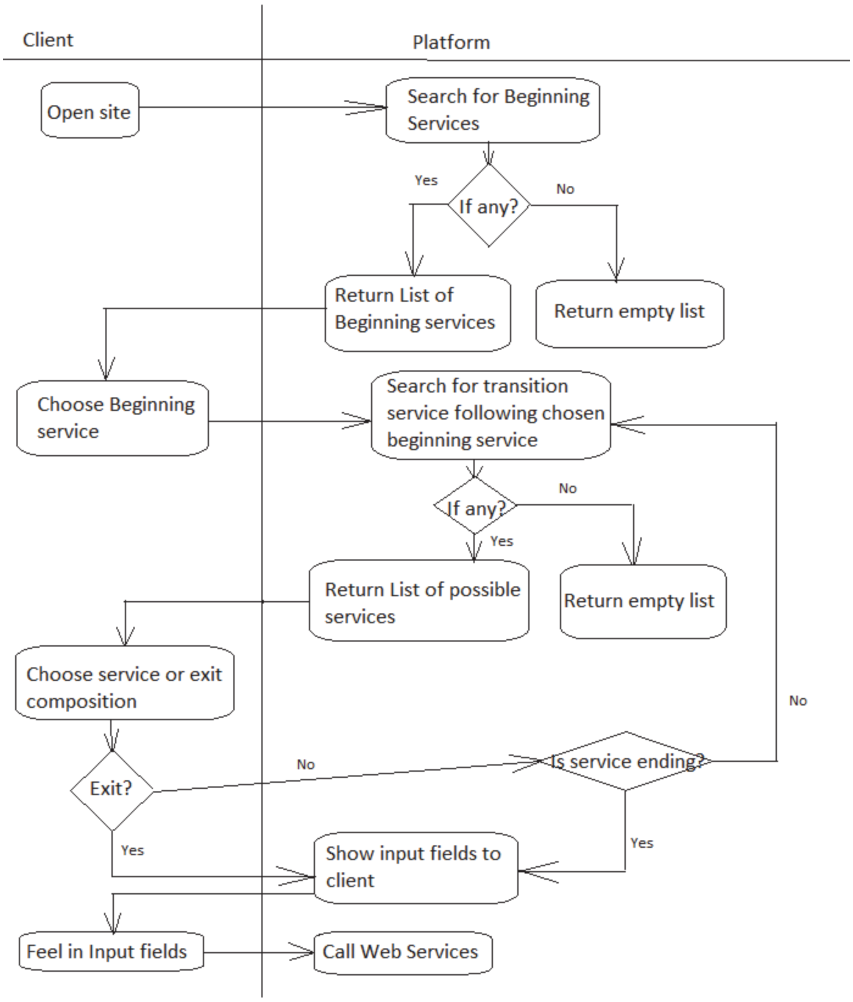
Client
Platform
Open site
Search for Beginning Services
If any?
Yes
No
Return List of Beginning services
Return empty list
Choose Beginning service
Search for transition service following chosen beginning service
If any?
No
Yes
Return List of possible services
Return empty list
Choose service or exit composition
No
Exit?
No
Is service ending?
Yes
Yes
Show input fields to client
Feel in Input fields
Call Web Services

Fig. 3. Flow of activities of the tool for interactive automatic composition of business processes

III. Tool for Automatic Service Composition

In this section, we present the design and implementation details about our automatic service composition tool. One of the most important tasks is to implement an algorithm, responsible for discovery of appropriate services. Main requirement for our tool is to propose recommendations to users about appropriate services for a composition. This is in order not only to automate the composition process, but also to help them to create their own business processes. For this purpose the tool should first implement service discovery rules. Afterwards, when appropriate services are found, users should select the services, they find most relevant to be included into the composition.

Additionally, one of our main quality requirements is about efficiency and performance of the tool. Simplified service discovery mechanism is implemented in order to enable service discovery and selection, while at the same time keeping efficiency and performance. Services are annotated with the two additional properties: service level and service category.

Service level marks the possible position of services into a composition. We have divided services into three levels: beginning, transition and ending services.

- Beginning services – these services may act only as a starting point of a business process. For example, these may be authentication services or services for searching a specific set of items.
- Transition services form the core of the business process – these are the services that may be called by a beginning service and they may call another transition service themselves. All transition services are additionally tagged with information about what other service (beginning or transition) may call them. This additional information is stored together with other service metadata.
- Ending service – are services that represent the final of a business process and are not supposed to result into call to another service. Ending services are also tagged with information about what other service (beginning or transition) may call them

Service level meta-information is usually set by administrators and sometimes during business process creation, users may also tag a transition service as ending if they want to finish their process.

However, in order for the algorithm for service recommendation to be fully functional, information about service level is not enough. For this purpose we define another meta-information tag about services, called category. Categories represent simplified description of the semantics of the service. This way, services may be matched by categories and users may select the category of next service in the business process. Afterwards, the tool selects the best service of this category to be included into the composition, based on QoS. These metadata about services are currently implemented into a database that acts as a service registry.

The tool implements interactive user communication, in which the user repeatedly selects which service wants to compose next. First the tool finds and lists to the user all available beginning services. After that based on metadata of the transition and ending services, all of them that may be called afterwards are listed. This is an iterative process, which ends with selection of a final service by the user (Fig. 3).

Composition of the selected services is implemented, following the orchestration approach. User decides to execute the service composition(represented by the "call services" state on Fig. 3), which causes the tool to read and parse the description of each participating service. This results into a web form, with input fields for all required data. The user should provide this data, as it is required by participating services for their execution, by filling the form. Finally the tool implements an orchestration approach that calls all selected services with the data provided by the user.

IV. Conclusion

Traditionally, composition of services in order to execute specific business processes required a certain amount of implementation and programming knowledge. However for the development of the IT society, such possibilities should be available for wider range of users with various computer literacy levels. This paper presents a tool that is capable to interactively facilitate users to build a composition of services into a business process. It also provides recommendation for appropriate services, this way facilitating creation of the business process.

This is an ongoing work and further research includes, improvement of the method for adding meta-information to services, as now this requires a significant amount of administrative work to tag all transition and ending services with list of services, allowed to execute them.

Acknowledgement

Research, presented in this paper was partially supported by the DFNI I02-2/2014 (ДФНИ И02-2/2014) project, funded by the National Science Fund, Ministry of Education and Science in Bulgaria.

References

[1] Z. Baida, J. Gordijn, and B. Omelayenko, "A Shared Service Terminology for Online Service Provisioning", *6th International Conference on Electronic Commerce*, Delft, The Netherlands, pp. 1-10, 2004.

[2] D. Claro, P. Albers, and J. Hao, "Web Services Composition", *Semantic Web Services, Processes and Applications*, Springer US, pp. 195-225, 2006.

[3] S. Dustdar, and W. Schreiner, "A Survey on Web Services Composition", *International Journal of Web and Grid Services* 1.1, pp. 1-30, 2005.

[4] T. Erl, *SOA: Principles of Service Design*, Pearson Education, Inc., Boston, USA, 2008.

[5] M. Hepp, et all, "Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management", *IEEE International Conference on e-Business Engineering*, pp. 535-540, 2005.

[6] M. Klusch, B. Fries, and K. Sycara, "Automated Semantic Web Service Discovery with OWLS-MX", *Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM, 2006.

[7] R. Nayak, and B. Lee, "Web Service Discovery with Additional Semantics and Clustering", *IEEE/WIC/ACM International Conference on Web Intelligence*, 2007.

[8] B. Orriëns, J. Yang, and M. Papazoglou, "Model Driven Service Composition", *International Conference on Service-Oriented Computing*, Springer Berlin Heidelberg, 2003.

[9] D. Petrova-Antonova, "A QoS-Aware Approach for Web Service Selection Based on Probability Evaluation", *M CC SSI MCCSIS 43*, 2011.

[10] D. Petrova-Antonova, and A. Dimov, "A QoS Driven Approach for Probability Evaluation of Web Service Compositions", *6th International Conference on Software and Database Technologies (ICSOFT)*, pp. 321-326, 2011.

[11] D. Petrova-Antonova, and A. Dimov, "Towards a Taxonomy of Web Service Composition Approaches", *Journal of Scalable Computing: Practice and Experience*, vol. 12, no. 4, pp. 377-384, 2011.

[12] I. Sommerville, *Software Engineering*, Addison Wesley, 2010.

[13] Web Services Business Process Execution Language Version 2.0, 2007, available at: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[14] OWL-S: Semantic Markup for Web Services, 2014. available at: http://www.w3.org/Submission/OWL-S/

[15] D2v1.4. Web Service Modeling Ontology (WSMO), 2007. available at http://www.wsmo.org/TR/d2/v1.4/

[16] Semantic Web Services Framework SWSF Version 1.0, 2005. Available http://www.daml.org/services/swsf/1.0/

[17] Cloudwork website, available at https://cloudwork.com

[18] WooFoo website, available at https://www.wufoo.com/

[19] If this than that website, available at https://ifttt.com/about

[20] Zapier website, available at https://zapier.com/