Variability Handling in Multi-Mode Service Composition

Peteris Rudzajs and Marite Kirikova

Institute for Applied Computer Systems
Riga Technical University
Riga, Latvia

ABSTRACT

Information services are designated for information processing intensive tasks and may require different levels of human involvement in their execution, e.g., in information processing and analysis. The handling of variability in information service systems incorporates the concern of human involvement in service execution or, as we denote it, functioning mode of service. The existence of different functioning modes of services raises a problem of multi-mode service composition. In the paper we propose the use and the extension of variability representation model to represent variability in the information service system and present the multi-mode service composition approach to derive consistent flows of both abstract and concrete services.

Keywords: Information service system, multi-mode services, feature modeling, service composition

INTRODUCTION

In this paper we consider information services and information service systems. Information service provides information and is "*a component of an information system representing a well defined business unit that offers capabilities to realize business activities and owns resources (data, rules, roles) to realize these capabilities*", whereas the information service system is a collection of interoperable information services (Ralyté, Khadraoui, & Léonard, 2013). In information service system the role of human actor is important because she/he can participate in service execution with different levels of involvement. To denote the level of human involvement, we use the term "functioning mode" of the service. We distinguish between the following three types of service functioning modes (Rudzajs, Kirikova, & Strazdina, 2013):

- *manual* - the service is performed by human actor (perhaps, using some office software, but there are no specific software services or tools included in the service system for implementing this service).
- *automatic* - the service is performed by dedicated software and/or hardware that does not require human actor intervention.
- *semi-automatic* - the service is performed by dedicated software and/or hardware that requires human involvement, e.g., a human performer should provide the input data and review and approve data processed and/or generated by the tool.

The term *multi-mode service* (Rudzajs et al., 2013) is used to characterize the service that can be instantiated in different functioning modes. The awareness and the use of different functioning modes in the service system raise

https://openaccess.cms-conferences.org/#/publications/book/978-1-4951-2091-6

the level of variability that should be handled when composing services. Variability in software engineering usually is defined as ability of software or software artifact (e.g. component) to be changed so that it fits a specific context (van Gurp, Bosch, & Svahnberg, 2001). As variability is an important factor in almost every contemporary system, many types of systems are built with the variability in mind. There are different types of variability, such as variability in features or in business processes (Galster, Avgeriou, & Tofan, 2013). The variability in features, particularly, variability in functioning modes of services is one of the concerns that should be considered in information service system. Not only services can be performed either by human performers (manually) or automatically, or semi-automatically; in some information service systems, one and the same abstract service can be instantiated in any of aforementioned functioning modes depending on the information handling situation. Currently variability handling is mainly considered in the context of single functioning mode of systems and their components. In this paper our goal is to focus on how to compose services with variable functioning modes.

The paper is organized as follows: The related work is briefly outlined in the next section. We propose the use and the extension of variability representation model to represent variability in the information service system and present the multi-mode service composition approach in the section "Functioning mode based variability of services". The extended variability representation model and multi-mode service composition approach are illustrated in the section "Example of extended feature model based service composition" using practical examples on education demand and offer monitoring service system (Rudzajs, 2012). Brief conclusions are stated in the last section of the paper.

RELATED WORK

Service composition involves the selection of appropriate services to satisfy particular user intention and can be defined as the process of combining and linking existing services (atomic or composite) to create new working services (Kapitsaki et al., 2007). Several surveys of service composition methods have been presented (Kapitsaki et al., 2007), (Regev, Favre, & Hayek, 2011), (Rao & Su, 2005), (Beek, Bucchiarone, & Gnesi, 2006), (Beek, Bucchiarone, & Gnesi, 2007), (Liu, Peng, Law, Wiederhold, & Sriram, 2005). The authors of aforementioned papers propose high variety of approaches that can be used for service composition. However, these approaches mainly consider different ways of composing application services with automatic functioning mode (Rudzajs et al., 2013). In this paper we are addressing various types of services from the point of view of their functioning modes. Variety of functioning modes poses the necessity to solve also the variability handling problem.

We propose to handle variability by (1) variability modeling and (2) multi-mode service composition approach that facilitate the inclusion of variability in information service systems. The proposed variability modeling approach is based on the following related work on variability in service systems and software engineering: (1) main variability research focus and application points, such as service variability modeling, service identification, service reuse, service configuration and customization, dynamic software product line, adaptive systems by Mohabbati et.al. (Mohabbati, Asadi, Gašević, Hatala, & Müller, 2013); (2) classification of variability in different dimensions that capture key facets of variability by Galster et.al. (Galster, Weyns, Tofan, Michalik, & Avgeriou, 2014). In Galster et.al. (2014) the dimensions of variability are organized in two clusters: type and mechanism. The type cluster includes dimensions for introduction and specification of variability, such as requirement type, representation, artifact, and orthogonality dimensions. The mechanisms of variability refer to the way the variability is realized. Our work considers the *Representation* dimension of the variability, namely, the application of service variability modeling and analysis by well known feature model (Kang, Cohen, Hess, Novak, & Peterson, 1990). Petersen et.al. (Petersen, Bramsiepe, & Pohl, 2006) propose model to support customer decisions by documenting alternatives in feature model and to communicate alternatives to a customer. In this paper we use the idea that feature models can serve as simple means to document and communicate alternatives of multi-mode service flows to particular stakeholders of the service system.

The main focus in the related work on variability handling is on services performed automatically by software components. Although some researches exist that concern several automation levels of services, e.g., (Sasa, Juric, & Krisper, 2008), (Sasa & Krisper, 2011), (Parasuraman, Sheridan, & Wickens, 2000), they do not consider the mix of different levels of automation. Also none of the authors discuss the services with different functioning modes. However the work of (Gu, Cuadrado, Lago, & Dueñas, 2013) propose three architectural viewpoints framing concerns about service automation. The viewpoints express architectural decisions about automation, identify the degree of automation, and represent specific data to support automation in services. In the model they illustrate service automation with different colors and identify which particular stakeholders are involved in particular service and what guides a stakeholder to provide the input for service. In the service composition they consider abstract services instead of concrete services realizing particular abstract service. In this paper we consider concrete services in the feature model and when composing multi-mode services. Thus our approach extends the approach presented in (Gu et al., 2013) by means of detailing service functioning mode to concrete services instead of just abstract services. This work also extends the related work (Gu et al., 2013) in a way that before composing the service flow we introduce the feature model that also can capture service automation concerns.

In the remainder of the paper we propose the composition of multi-mode services based on variability representation model extended with the service functioning modes.

FUNCTIONING MODE BASED VARIABILITY OF SERVICES

We discuss the functioning modes of services in more detail and describe how they can be represented in the feature model in the first subsection of this section. The second subsection is devoted to the description of multi-mode service composition approach. The third subsection discuses some features of the proposed approaches of variability modeling and service composition.

Extending Feature Model

Well known feature models show the variability in service systems (Wittern & Zirpins, 2011). However, they do not address directly the issue of functioning modes of services. Therefore the feature model has to be extended to accommodate aforementioned variability of functioning modes. To achieve this, in the model we represent services as features provided by the information service system. Other representation types also could be considered (e.g., ontologies (Lamprecht, Naujokat, & Schaefer, 2013)), but they are out of the scope of this paper. In the feature model we distinguish between two types of services, namely, *abstract services* (represented as variation points in rectangular boxes in Figure 1) and *concrete services* (represented as variants in boxes with rounded corners in Figure 1). By variation point we denote a particular place in a system where choices are made as to which variant to use (Svahnberg, van Gurp, & Bosch, 2005). Variant is a particular option of a variation point. These concrete services are supposed to implement abstract services. We use the abbreviation *AS* for abstract services and *CS* for concrete services. An abstract example of feature model consisting of one abstract service (variation point) and three concrete services (variants) is presented in left part of Figure 1.

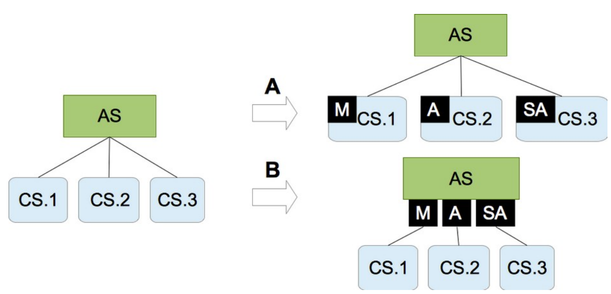

Figure 1. Feature model extended with the service functioning mode. Functioning modes are represented as A (automatic), M (manual), and SA (semi-automatic).

We discuss two options of feature model representation concerning the functioning mode of service (see the right part of Figure 1). In the first option the functioning mode is directly captured in concrete service as a parameter (see part A of Figure 1). Whereas in the second option the functioning mode is captured at an additional layer of abstract service (see part B of Figure 1). Main difference is in how the feature model is structured. First option does not introduce new layer for abstract services. In this case it becomes harder to manage the model when the variability of concrete services and their functioning modes grows. In the second option the functioning modes of abstract service are grouped within the additional layer thus making the feature model more structured and manageable when growing. With respect to this concern, further in the paper we use the second option to represent the variability of functioning modes in the feature model. Within this option each concrete service should be reviewed and assigned to particular group of functioning mode (e.g., the concrete service CS.1 is assigned to manual functioning mode of abstract service AS in part B of Figure 1).

Essential benefits from the feature model extension by functioning modes are the possibility to clearly identify the services where human involvement is necessary (manual and semi-automatic functioning modes) even before the composition of services and the possibility to identify the alignment of interfaces when composing multi-mode services.

Composition of multi-mode services

In this subsection we propose an approach to identify consistent compositions (service flows) of abstract and concrete services based on extended feature model.

Availability of the property of functioning mode and "required by" relationships in the feature model provide basis for further decisions to ensure consistent service flow. While the feature model itself lacks the ability to represent the composition of services (service flow), since this is not the aim of the feature model, still, the feature model with depicted service functioning modes could serve as a solid basis for composing multi-mode services. An abstract example of feature model consisting of four abstract services AS (variation points) and eight concrete services CS (variants) for three abstract services is presented in Figure 2. We will use this example to discuss the steps of multi-mode service composition (MMSC) approach presented further in this section. Figure 2 illustrates the following types of "required by" relationship (see corresponding numbers in circles): (1) between abstract services, (2) between abstract service and concrete service, (3) between concrete services.

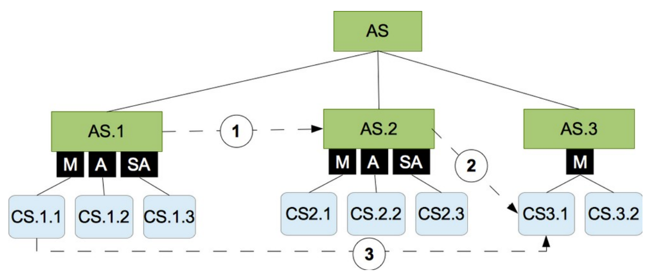

Figure 2. An abstract example of feature model extended with functioning modes of services. Abstract services (AS) are represented in rectangular boxes, concrete services (CS) – in boxes with rounded corners. Functioning modes are represented as additional layer for abstract services as A (automatic), M (manual), SA (semi-automatic). Dotted arrows are used for "required by" relationship.

By service composition we mean the arranging of services in a particular service flow. The service flow indicates the sequence of execution of services that is prescribed at the design time. Multi-mode service composition (MMSC) approach has to be performed in the following steps:

1) *Based on the intention user wants to achieve, compose the flow(s) of abstract services.* In the feature model, the variants of abstract services are either concrete services or other abstract services. When designing the flow of abstract services we should choose only abstract services for which the variants are concrete services (with particular functioning mode). We call abstract services, for which the variants are other abstract services, as *grouping services* and consider them only when checking the service flow for consistency (e.g., in the 1.b and 1.c sub-steps below). Consistent service flow is the service flow composed of services (abstract or concrete) satisfying the "required by" relationships given at the feature model. The flow of abstract services is composed by the following sub-steps:
   a) Choose initial abstract services from the feature model and arrange them in the flow.
   b) For each abstract service check *directly required* abstract services (e.g., AS.2 directly requires AS.1, indicated as (1) in Figure 2) or grouping services and update the service flow with the new services if necessary.
   c) For each abstract service check *indirectly required* abstract services. Indirectly required services can be determined in the following way: for each concrete service of the abstract service check directly required 1) abstract services, e.g., CS.3.1 directly requires AS.2, indicated as (2) in Figure 2; 2) concrete services, e.g., CS.3.1 directly requires CS.1.1, indicated as (3) in Figure 2; or 3) grouping services. Include appropriate

abstract services as “optional” in the flow if they are required by at least one concrete service. In the current example, AS.2 is included in the flow as optional abstract service (see part A of Figure 3).

2) *Derive the service flows of concrete services* by the following sub-steps:
   a) Choose concrete service for each abstract service in the initial flow.
   b) For chosen concrete services include directly (e.g., (3) in Figure 2) and indirectly (e.g., (2) in Figure 2) required concrete services and mark appropriate optional abstract service as required (e.g., see flow 1 in right hand side of part B of Figure 3).
   c) Repeat 2.a) and 2.b) sub-steps until all eligible concrete services are arranged in flows
   d) Arrange the flows by adding the flows with the initial abstract services at the beginning of the list. For the abstract flow in this particular example we have 3 options for concrete flows with 2 initial services and 9 options for concrete flows of 3 services. Concrete flows are illustrated in part B of Figure 3.
3) *Identify stakeholders involved in the service flows*. Both M and SA functioning modes of service require the involvement of human actors in the execution of service.

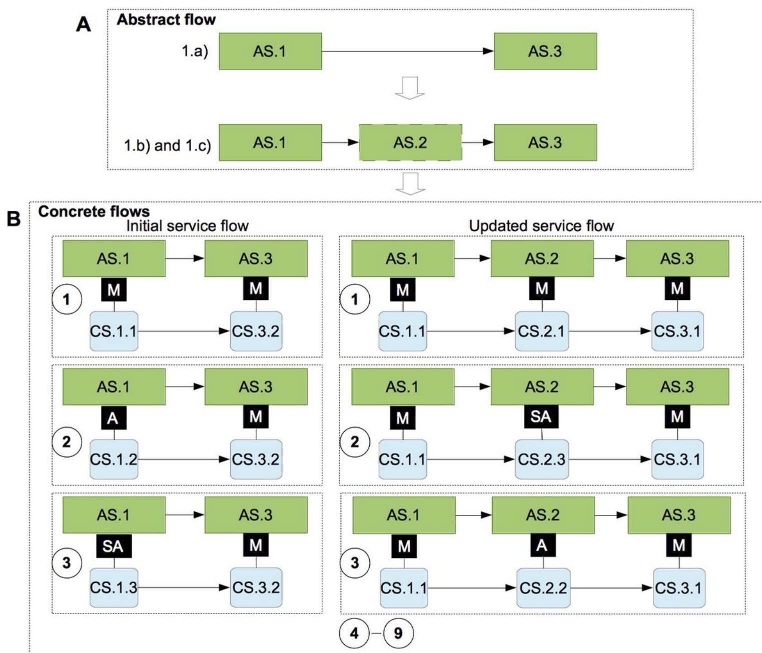

Figure 3. MMSC approach illustrated on an abstract example. Part A of the figure illustrates the 1.a) to 1.c) steps of the approach. Part B of figure illustrates the flows of concrete services organized by the initial service flow and the updated service flow (includes *optional* service AS.2).

When dealing with multi-mode services, the alignment of service interfaces becomes one of the main concerns to enable service interaction. Typically, in service oriented systems the alignment is done between services with automatic functioning mode only. In multi-mode information service systems due to variety of functioning modes the number of possible interfaces increases. This highlights the variability issue in the alignment of service interfaces. The following alignment possibilities of (multiple) input and output interfaces or communication possibilities can be designated or provided for passing and retrieving information to/from concrete service with particular functioning modes:

- *From automatic service to automatic service.* Application level service interfaces should be aligned between services in composition. Human actor is not involved.
- *From automatic service to manual service.* User interface should be established allowing particular stakeholder of service with manual functioning mode to review the output of service with automatic functioning mode. Human actor is involved.
- *From manual service to automatic service.* User interface should be established (usually as input forms) allowing particular stakeholder the preparation of the result of service with manual functioning mode for input into service with automatic functioning mode.

https://openaccess.cms-conferences.org/#/publications/book/978-1-4951-2091-6

- *From manual service to manual service*. Specific application level interfaces are required (usually as input forms); business level communication is possible.
- *SA involved*. Depending on the specifics of the service with semi-automatic functioning mode, it may require only application level interface, only business level communication, or both.

By taking into account the alignment of service interfaces, we extend the MMSC approach to revisit the concrete service flows derived by steps 1, 2 and 3 of the MMSC approach by considering the applicable transitions between services with the same or different functioning modes. Thus we add step 4 to the MMSC approach:

4) *Revisit the options of concrete service flows* by considering the alignment of interfaces necessary in composition of two services and omit the unwanted/not reliable alignment of interfaces between services, thus omitting also the particular flows of concrete services.

Remaining flows of concrete services should be considered for implementation in information service system.

Discussion

The above-presented MMSC approach currently gives an opportunity to utilize the variability clearly stated in the extended feature model. The following issues are important for manual composition of services:

- MMSC approach mainly uses “required by” relationship to derive consistent flow of abstract and concrete services. “Required by” relation is important in the feature model because it allows finding appropriate flows of concrete services, as well it allows crosschecking the abstract service flow for consistency.
- Human actor determines the sequence of services in all cases. MMSC approach allows to compose consistent flow by using 1.b) and 1.c) steps and it allows to reason about the alignment of interfaces involved in the service flow, that could be considered for design and implementation of the information service system where the variability in functioning modes of services exists. Service flow could be implemented in software system, i.e., software system can initiate the services in the particular sequence.
- Certainly, the feature model can change during design and evolvement of the information service system. Each change in the feature model should be reflected also in the flows of abstract and concrete services to maintain consistent service flows.

To be able to automatically derive consistent service flows with multimode services, the feature model should be extended with algorithms employing formal rules. This is one of our future research goals.

EXAMPLE OF EXTENDED FEATURE MODEL BASED SERVICE COMPOSITION

This section presents the feature model and apply MMSC approach for education demand and offer monitoring system (EduMON) (Rudzajs, 2012).

Description of Functional Variability in EduMON

EduMON is information service system for information handling with respect to different information sources and stakeholders. It is aimed at supporting education demand and offer monitoring process by the following activities (feature model of EduMON is provided in Figure 4):

- *Providing* activity is for providing documents from available reachable information sources. The documents available in information sources should reflect the information about demanded and offered knowledge, skills, and competences and are retrieved from different types of textual sources (e.g., Web sites, databases, XML-based files).
- *Processing* activity aims at extracting education information (knowledge, skills, and competences) from the information sources (particularly, from the documents) available in the system and, by comparing information from different sources, to depict the education demand and offer correspondence.

- *Consuming* activity distributes the processed information to the stakeholders of the system via graphical and tabular reports.

Various stakeholders (e.g., teachers, students, employers, and others) are involved in each above-mentioned activity. They are grouped with respect to the activity they are involved, namely, *Providers*, *Processers* and *Consumers*. Stakeholders interact via, with, and within EduMON to fulfill specific information handling intentions.

Providing and *Processing* activities are targeted to particular information sources and their documents (represented in part I of Figure 4), however, the *Consuming* activity is for representing the processed information from multiple information sources to users via graphical and tabular reports (part II of in Figure 4). In the feature model provided in Figure 4, solid lines represent the types (mandatory or optional) and relationships (OR or XOR) of abstract services. Dashed lines represent the "required by" relationship between abstract and/or concrete services. For instance, to execute any of the concrete services of *Extraction* abstract service (see 2.1 in Figure 4), the execution of any of the concrete services of *Retrieval* abstract service (see 1.2 in Figure 4) is required.

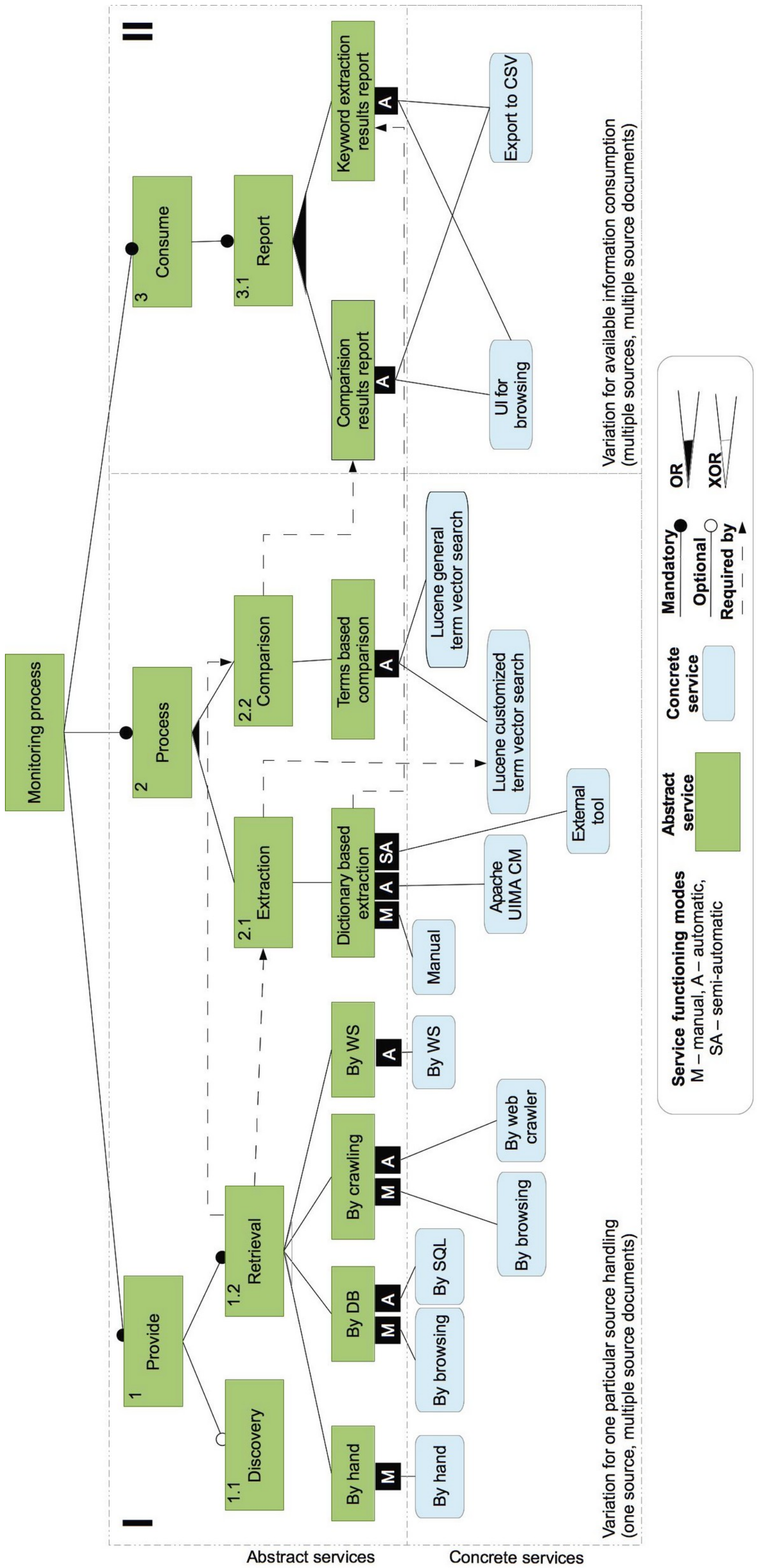

Figure 4. The functional variability of EduMON represented by the feature model extended with functioning modes of abstract services and their concrete implementations.

Multi-Mode Service Composition in EduMON

In this subsection we discuss the application of MMSC approach to EduMON information service system.

1) First we compose the flow of abstract services. Suppose, our intention is to add new information source by hand and perform its comparison against other information sources in EduMON (e.g., add new curriculum and compare it with other curricula to find similar courses).
    a) Initial service flow includes two abstract services, namely, *Retrieval by hand* and *Terms based comparison* (see part A of Figure 5).
    b) There are no directly required abstract services for *Retrieval by hand* and *Terms based comparison*, thus the service flow is not updated at this sub-step.
    c) For the concrete service *Lucene customized term vector search* directly required abstract service can be identified, namely, *Dictionary based extraction* (see Figure 4). Thus we update abstract service flow and include *Dictionary based extraction* as optional abstract service (see part A of Figure 5), because it is required by one concrete service of *Terms based comparison* abstract service.
2) Now we derive the flows of concrete services to achieve the intention stated in Step 1 (see part B of Figure 5). There is one option for initial flow of abstract services and three options for updated flow of abstract services.
3) The stakeholders involved in each particular flow of concrete services belong to stakeholders of *Providing* and *Processing* activity groups. The *Provider* stakeholder group is responsible for the *Retrieval by hand* service with manual functioning mode, whereas the *Processer* stakeholder group - for the *Dictionary based extraction* service with manual and semi-automatic functioning mode. As *Terms based comparison* service is performed with automatic functioning mode the involvement of human actor is not necessary. Discussion about the groups of stakeholders is available in (Rudzajs et al., 2013).
4) By reviewing the options of concrete service flows and considering the alignment of interfaces necessary in the composition of two services, we conclude that all flows are feasible to be implemented in the system. For instance, the service *Retrieval by hand* with manual functioning mode is in composition with the service *Dictionary based extraction by Apache UIMA ConceptMapper* with automatic functioning mode (see the first updated flow of concrete services in part B of Figure 5). In this case we should establish interface to allow the passing the retrieved document from service *Retrieval by hand* to the service *Dictionary based extraction by Apache UIMA ConceptMapper*. This interface can be implemented as input form where the human actor passes the document, e.g., study course description, to automatic keyword extraction service.

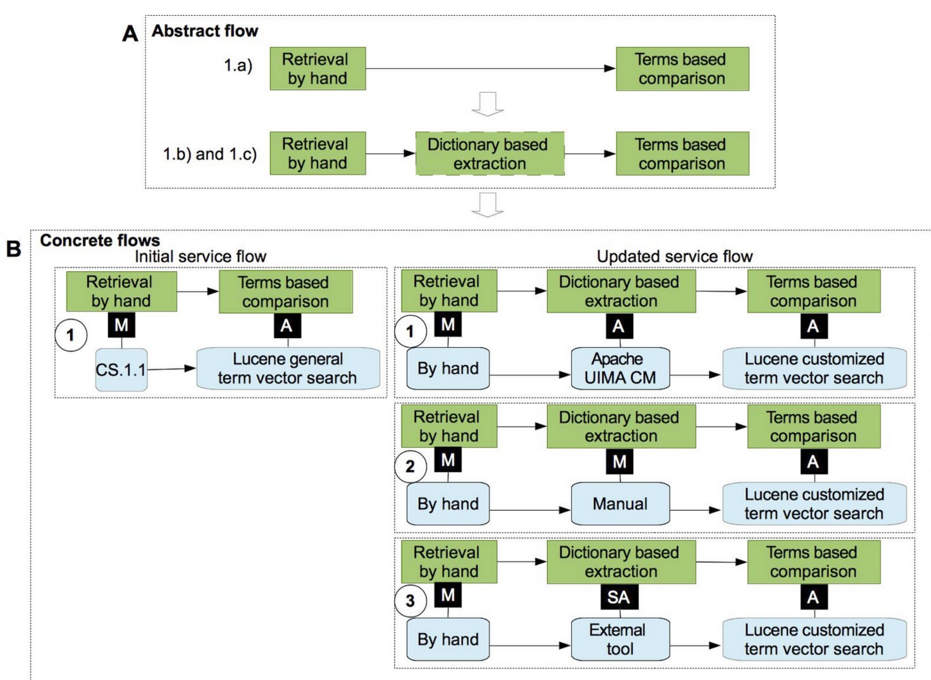

Figure 5. EduMON multi-mode service composition for intension “add new information source by hand and perform its

comparison to other information sources".

The application of the MMSC approach to EduMON information service system enabled us to analyze the system from the viewpoint of variability, thus allowing to derive potential usage scenarios involving the services with different functioning modes. Because of relatively few "required by" relationships in the feature model (see Figure 4), the composition of multi-mode services were quite straightforward; however, for systems with greater functional variability the automated approach might be more feasible.

CONCLUSIONS

In this paper we approached the problem of variability handling in multi-mode service composition. To solve the problem, we suggested to extend the well known feature model for service representation and proposed the approach of service composition (MMSC approach) that is based on the extended feature model. With the extension the feature model becomes more expressive since it reflects the concern about functioning modes of services in information service system. We showed two options how the feature model can be extended, namely, by assigning the functioning mode directly to the concrete services or by assigning concrete services to particular group of functioning mode of abstract service. Second option provides more structured and manageable feature model. The extended feature model provides the possibility to clearly identify the services where the human involvement is necessary even before the actual composition of services; as well it serves as the ground for MMSC approach to derive consistent flows of both abstract and concrete services. Application of extended feature model and the MMSC approach facilitates the handling of variability in information services systems where the variability of functioning modes of services is present.

Further research direction includes the extension of feature model with formal rules allowing automatic derivation and maintenance of consistent service flows with multi-mode services.

REFERENCES

Beek, M. Ter, Bucchiarone, A., & Gnesi, S. (2006). *A survey on service composition approaches: From industrial standards to formal methods*. In Proceedings of the 2nd International Conference on Internet and Web Applications and Services (ICIW'07).

Beek, M. Ter, Bucchiarone, A., & Gnesi, S. (2007). *Formal methods for service composition*. Annals of Mathematics, Computing & Teleinformatics, *1*(5), 1–10.

Galster, M., Avgeriou, P., & Tofan, D. (2013). *Constraints for the design of variability-intensive service-oriented reference architectures – An industrial case study*. Information and Software Technology, *55*(2), 428–441. doi:http://dx.doi.org/10.1016/j.infsof.2012.09.011

Galster, M., Weyns, D., Tofan, D., Michalik, B., & Avgeriou, P. (2014). *Variability in Software Systems - A Systematic Literature Review*. IEEE Transactions on Software Engineering. doi:10.1109/TSE.2013.56

Gu, Q., Cuadrado, F., Lago, P., & Dueñas, J. C. (2013). *3D architecture viewpoints on service automation*. Journal of Systems and Software, *86*(5), 1307–1322. doi:10.1016/j.jss.2012.12.035

Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). *Feature-oriented domain analysis (FODA) feasibility study*.

Kapitsaki, G., Kateros, D., Foukarakis, I., Prezerakos, G., Kaklamani, D., & Venieris, I. (2007). *Service Composition: State of the art and future challenges*. In Proceedings of the Mobile and Wireless Communications Summit, 2007 (pp. 1–5).

Lamprecht, A.-L., Naujokat, S., & Schaefer, I. (2013). *Variability Management beyond Feature Models*. Computer, *46*(11), 48–54. doi:http://doi.ieeecomputersociety.org/10.1109/MC.2013.299

Liu, D., Peng, J., Law, K., Wiederhold, G., & Sriram, R. (2005). *Composition of engineering web services with distributed data-flows and computations*. Advanced Engineering Informatics, *19*, 25–42.

Mohabbati, B., Asadi, M., Gašević, D., Hatala, M., & Müller, H. A. (2013). *Combining service-orientation and software product line engineering: A systematic mapping study*. Information and Software Technology, *55*(11), 1845–1859. doi:http://dx.doi.org/10.1016/j.infsof.2013.05.006

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). *A model for types and levels of human interaction with automation.* IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society, *30*(3), 286–97.

Petersen, K., Bramsiepe, N., & Pohl, K. (2006). *Applying Variability Modeling Concepts to Support Decision Making for Service Composition*. In Service-Oriented Computing: Consequences for Engineering Requirements, 2006. SOCCER '06 (p. 1). doi:10.1109/SOCCER.2006.1

Ralyté, J., Khadraoui, A., & Léonard, M. (2013). *From Information Systems to Information Services Systems: Designing the*

Transformation. In J. Grabis, M. Kirikova, J. Zdravkovic, & J. Stirna (Eds.), The Practice of Enterprise Modeling SE - 6 (Vol. 165, pp. 69–84). Springer Berlin Heidelberg. doi:10.1007/978-3-642-41641-5_6

Rao, J., & Su, X. (2005). *A survey of automated web service composition methods*. In Proceedings of SWSWPC 2004 (pp. 43–54).

Regev, G., Favre, J., & Hayek, E. (2011). *Business/IT Alignment in Practice: Lessons Learned from a Requirements Project at P&G*. In Advanced Information Systems Engineering Workshops (pp. 93–101).

Rudzajs, P. (2012). *Towards Automated Education Demand-Offer Information Monitoring: the System's Architecture*. In Selected Papers from Workshops and Doctoral Consortium of the International Conference on Perspectives in Business Informatics Research, BIR2011 (pp. 252–265). Springer, Heidelberg.

Rudzajs, P., Kirikova, M., & Strazdina, R. (2013). *Configurative Alignment of Business and Application Services: A Work Systems Perspective*. In Lecture Notes in Business Information Processing (pp. 100–111). Springer Berlin / Heidelberg. doi:10.1007/978-3-642-41687-3_11

Sasa, A., Juric, M. B., & Krisper, M. (2008). *Service-Oriented Framework for Human Task Support and Automation*. IEEE Transactions on Industrial Informatics, *4*(4), 292–302. doi:10.1109/TII.2008.2008641

Sasa, A., & Krisper, M. (2011). *Service-oriented architectural framework for support and automation of collaboration tasks*. Journal of Information and Organizational Sciences, *35*(1), 119–133.

Svahnberg, M., van Gurp, J., & Bosch, J. (2005). *A Taxonomy of Variability Realization Techniques: Research Articles*. Softw. Pract. Exper., *35*(8), 705–754. doi:10.1002/spe.v35:8

Van Gurp, J., Bosch, J., & Svahnberg, M. (2001). *On the notion of variability in software product lines*. In Proceedings of the Working IEEE/IFIP Conference on Software Architecture (pp. 45–54). doi:10.1109/WICSA.2001.948406

Wittern, E., & Zirpins, C. (2011). *On the use of feature models for service design: The case of value representation*. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6569 LNCS, pp. 110–118).