Service Functioning Mode in Variability Model

Peteris Rudzajs and Marite Kirikova

Institute of Applied Computer Systems, Riga Technical University, Latvia
{peteris.rudzajs,marite.kirikova}@rtu.lv

Abstract. Recently variability handling has become a very important research topic due to necessity to provide higher flexibility in business and software operations. Usually variability is discussed either at business operations level or at software operations level. However, often both types of operations must be taken into consideration, especially in information intensive tasks, where human actors as well as computer systems are handling the information. Information intensive tasks are common in information service systems. Therefore description and use of variability from information handling perspective is important when designing and implementing this type of systems. In the paper we consider variability in the context of information services and information service systems. The paper proposes extended feature model based approach for capturing key variability facets in information service systems. Practical application of the approach is illustrated by the education demand and offer monitoring service system.

Keywords: variability, multi-mode service, information service, information service system, monitoring system.

1 Introduction

Variability is the main factor in almost every system [1]. Many types of systems are built with the variability in mind, e.g., self-adaptive systems, open platforms, and service-oriented systems. Variability handling can be supported by different variability management tools, software configuration wizards and tools, software component configuration interfaces, as well as by infrastructure for dynamic service composition [1].

In this paper we consider variability in the context of information services and information service systems. Information service [2] 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. In software based information service system a human actor of an information service has to be considered because it can participate in service execution with different degrees of involvement up to the degree where only the human actor performs the service. We use concept "functioning mode of service" to denote the degree of human involvement in service execution.

L. Iliadis, M. Papazoglou, and K. Pohl (Eds.): CAiSE 2014 Workshops, LNBIP 178, pp. 124–135, 2014.

There are different types of variability, such as variability in features or in business processes [3]. In an information service system the variability in features, particularly, variability in functioning modes of services is one of the concerns that should be considered, since the services can be performed only by human actors (manually), automatically, or semi-automatically. Moreover, 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 most of variability models are designed to handle variability for systems and their components of single functioning mode. In this paper our goal is to focus on differences of functioning modes of services and analyze the impact of these differences on the variability representation. For variability representation we use well known feature model [4] to document and analyze the mandatory, optional, and alternative features of the system and to communicate them to stakeholders of the system. In feature model we represent services as features provided by the information service system. Our contribution in this paper is an approach to introduce functioning modes of services in the feature model that facilitates design and implementation of the information service system.

The paper is organized as follows: In Section 2, the basic concepts used in the paper are described and related work is briefly outlined. In Section 3, the approach for extending the feature model by assigning functioning mode properties to the services is proposed. In Section 4, extended feature model is discussed using practical example of education demand and offer monitoring service system [6]. Brief conclusions are stated in Section 5.

2 Basic Concepts and Related Work

Basic concepts used in the paper and related work are briefly outlined in the following subsections.

2.1 Basic Concepts

The following basic concepts are used in the paper:

- *Information service* [2] is a component of an information system representing a well defined business unit. This unit offers capabilities to realize business activities and owns resources (data, rules, roles) to realize these capabilities. We assume that the "business unit" here can own human performers only, artificial performers (software and hardware components) only, or both human and artificial performers.
- *Information service system* [2] is a collection of interoperable information services.
- *Variability* – in software engineering the variability usually is defined as ability of software or software artifact (e.g. component) to be changed so that it fits a specific context [7]; here we take an information handling perspective and define the variability as ability to change the information handling unit so that it fits a specific context, goal, or intention.

- *Variation point* denotes a particular place in a system where choices are made as to which variant to use [8].
- Variant is a particular option of a variation point [8].
- *Functioning mode* of service [9]: *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.
- *Multi-mode service* (or service with mode variation) [9] - service that can be instantiated in different functioning modes.

2.2 Related Work

The approach of variability modeling discussed in this paper is based on related work on variability in service systems and software engineering. Mohabbati et.al. [10] identify the main variability research focus and its application points, namely, *service variability modeling*, service identification, service reuse, service configuration and customization, dynamic software product line, and adaptive systems. Galster et.al. [1] provide the classification of variability in different dimensions that capture key facets of variability. Classification can be used as the baseline from which the key aspects of variability of different types of software systems can be identified and compared. Galster et.al. [1] identify also the dimensions of variability that are organized in two clusters namely, the type and the mechanism. The type cluster includes *dimensions for introduction and specification of variability*, namely, requirement type, representation, artifact, and orthogonality dimensions. The mechanisms cluster of variability refers to the way variability is realized. Our work considers variability modeling [10] and the *representation* dimension for introduction and specification of variability [1]. For variability representation we use well known feature model [4]. Feature model is used to document and analyze the mandatory, optional, and alternative features of system and to communicate them to the stakeholders of the system.

Lamprecht et.al. [11] look at the variation in processes and provide variation realization approach to automatically implement and manage concrete process variants. All variants are described by means of domain model (consisting of services, ontologies, and constraints) and constructed by means of a synthesis algorithm. From this work we use the idea of constraining available (possible) process variants and extend it with respect to the automation of process variant generation in cases where multiple functioning modes of services are available from the information handling perspective.

Stollberg and Muth [12] propose method for service customization, by using model driven variability management. Service variability aspects (mandatory and optional operations, properties of message types and their dependencies) are described on the meta-model basis. Consistent variants are derived depicting only those features that are important for the customer. Meta-model design including the details of functioning modes of services could be used in our future work, but is not introduced in this paper.

Petersen et.al. [13] propose the model to support customer decisions by documenting alternatives in the feature model and to communicate alternatives to the customer. In this paper we use the idea that feature models can serve as simple means to document and communicate alternatives to particular stakeholders of the service system. We use the feature models to (1) document variability aspect in information service system, (2) to analyze the potential human involvement and service interfaces for particular functioning mode, and (3) to draw further implementation considerations.

Nguyen and Colman [14] propose feature-oriented approach for web service customization addressing three main challenges: reducing complexity, automatic validation and dynamic deployment. Authors of [15] and [16] provide six variability patterns for service oriented computing domain that can guide developers to solve different variability problems in practice. Patterns include Parameter Pattern, Routing Pattern, Service Wrapping Pattern, Variant/Template Pattern, Extension Points Pattern, Copy, and Adapt Pattern. Authors of [14], [15], and [16] focus only on automatic web services. In the context of information service systems we should consider services with other functioning modes, too; such as services performed manually or services performed semi-automatically by support of external tools or systems.

In general, we can conclude that in the related work the main focus is on services performed automatically by software components. There exist some researches that concern several automation levels of services, e.g., [17], [18], and [19], but they do not consider the mix of different levels of automation. Also none of the authors discuss the variation points and variants with different functioning modes.

In the remainder of the paper we will examine how consideration of service functioning modes can impact the variability representation model.

3 Functioning Modes in Feature Model

In this section we depict the functioning modes of services and describe how these modes could be represented. We use the feature model to show the variability in information service system (the initial design of the feature model is out of the scope of this paper). In the model we represent services as features provided by the information service system. Other representation types also could be considered (e.g., ontologies [11]), 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 Fig. 1) and *concrete services* (represented as boxes with rounded corners in Fig. 1). 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 part A of Fig. 1. We propose to assign a particular functioning mode to a particular concrete service as a property using functioning mode assignment (FMA) approach. The FMA approach uses the feature models and prescribes the following steps for functioning mode assignment (practical illustration of the approach is given in Section 4):

1. In the given feature model, review concrete services (see part A in Fig. 1) of each abstract service.
2. Add the property of functioning mode to each concrete service, if it cannot be instantiated in any other functioning mode (see part B in Fig. 1; for concrete services CS.1.1 and CS.1.3 the property of functioning mode is added, namely, for CS.1.1 functioning mode is automatic - A, for CS.1.3 it is manual – M).
3. Add variability to each concrete service by converting it into abstract service and identify new concrete services, if the concrete service (variant) can be instantiated in more than one functioning modes; and repeat Step 2. In our abstract example CS.1.2 can be instantiated in 2 functioning modes (see CS.1.2 in part A and AS.1.2 in part B in Fig. 1). After repeating Step 2, we have added functioning modes to CS.1.2.1 (automatic - A) and to CS.1.2.1 (semi-automatic - SA).
4. After the functioning modes are added to concrete services, interfaces designated for transition between services with different or the same functioning modes should be added (see part C in Fig. 1).

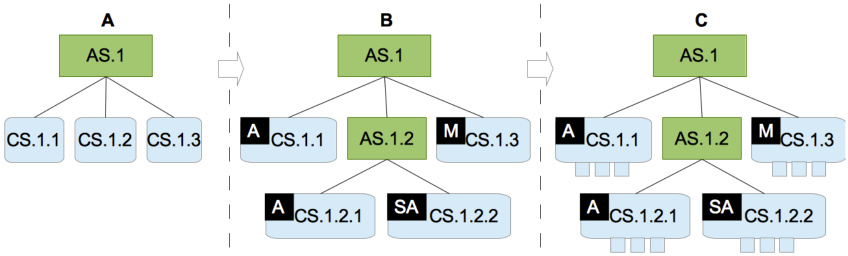

Fig. 1. Adding variability with respect to functioning modes of services. Abstract services (AS) are represented in rectangular boxes, concrete services (CS) – in boxes with rounded corners. Functioning modes are represented in concrete services as A (automatic), M (manual), SA (semi-automatic).

Since in information handling we should consider the functioning modes of concrete services, the representation of these modes will help designers or implementers of a service system to determine the types of service interfaces necessary for transition between services with the same or different functioning modes. This highlights the necessity to consider the variability in service interfaces. In Table 1 we briefly illustrate the need for (multiple) input and output interfaces or communication possibilities to be designated or provided for passing and retrieving information to/from services with particular functioning modes. The following transitions between services are considered A-A, A-M, M-A, M-M and any combinations where SA mode is involved. For each concrete service there should be considered three interfaces for transition to each of the possible functioning modes of the service (see concrete services in part C in Fig. 1).

Table 1. Interfaces between services of various functioning modes (A (automatic), M (manual), or SA (semi-automatic)) (adapted from [9])

Functioning modes	**Interface involved**	**Human actor involved**
A-A	Application level service interfaces should be established between services in transition	-
A-M	User interface should be established allowing particular stakeholder of service with manual (M) functioning mode to review the output of service with automatic (A) functioning mode	+
M-A	User interface should be established (usually as input forms) allowing the preparation of the result of service with manual (M) functioning mode for input into service with automatic (A) functioning mode	+
M-M	Specific application level interfaces are required (usually as input forms), business level communication could be possible	+
SA involved	Depending on the specifics of the service with semi-automatic (SA) functioning mode, it may require only application level interface, only business level communication, or both.	+

FMA approach allows adding the functioning mode for concrete services. Availability of this property provides basis for further decisions with respect to interfaces necessary in information service system for transitions between the services (to ensure proper 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 developing other models aimed at composing multi-mode services (see identified compositions for a particular example in Table 2).

Information service system can evolve; therefore it is necessary to deal with changes of features provided by the system. Potential changes include adding, removing, and updating of abstract and concrete services. When planning the changes of the system we should reflect the changes in the feature model and analyze the impact on the service compositions and service flows already available in the system. For instance, in case of adding new service to the system, (1) the feature model should be updated by adding this service to the model and (2) FMA approach should be applied to the service to assign to the functioning mode (or add variability to concrete service by converting it to abstract service and identifying new concrete services) and to assign the interfaces to it. We consider in this paper the assignment of functioning modes at the level of concrete services, however it could be done also at the level of abstract services. Then it would require another way for feature model change management, as well as the extension of FMA approach. The construction and examination of such feature model is one of our further research directions.

4 Practical Example and Discussion

In this section we define the feature model and apply FMA approach for education demand and offer monitoring system (EduMON) [6].

EduMON is information service system for information handling with respect to different information sources and stakeholders. EduMON service system is aimed at supporting education demand and offer monitoring process by the following activities (feature model of EduMON is provided in Fig. 2):

- *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.

In each above-mentioned activity, various stakeholders (e.g., teachers, students, employers, and others) are involved. 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 Fig. 2), however the *Consuming* activity is for representing the processed information from multiple information sources to users via graphical and tabular reports (part II of Fig. 2). In the feature model provided in Fig. 2, solid lines represent the types (mandatory or optional) and relationships (OR or XOR) of abstract services. Dashed lines represent “required by” relationship between abstract and/or concrete services. For instance, to execute any of the concrete services of *Extraction* abstract service (see 1.2 in Fig. 2), the execution of any of the concrete services of *Retrieval* abstract service is required.

In Fig. 3 we provide the feature model extended by FMA approach proposed in Section 3. First, we review the initial feature model provided in Fig. 2 (Step 1 of the FMA approach). By following Step 2 of the approach, 10 out of 12 initial concrete services were updated and additional property characterizing the functioning mode was added.

By following Step 3 of the approach, 2 out of 12 services showed additional variability in functioning modes; therefore these two initially concrete services were converted to abstract services (namely variants *Retrieval by database* 1.2.2. and *Retrieval by crawling* 1.2.3). For each of new abstract services two concrete services were introduced and additional property characterizing their functioning mode was added. *Retrieval by database* (1.2.2) variant was converted to variation point and two variants were added, namely, by browsing database manually (1.2.2.1) and by using SQL to automatically retrieve data from the database (1.2.2.2).

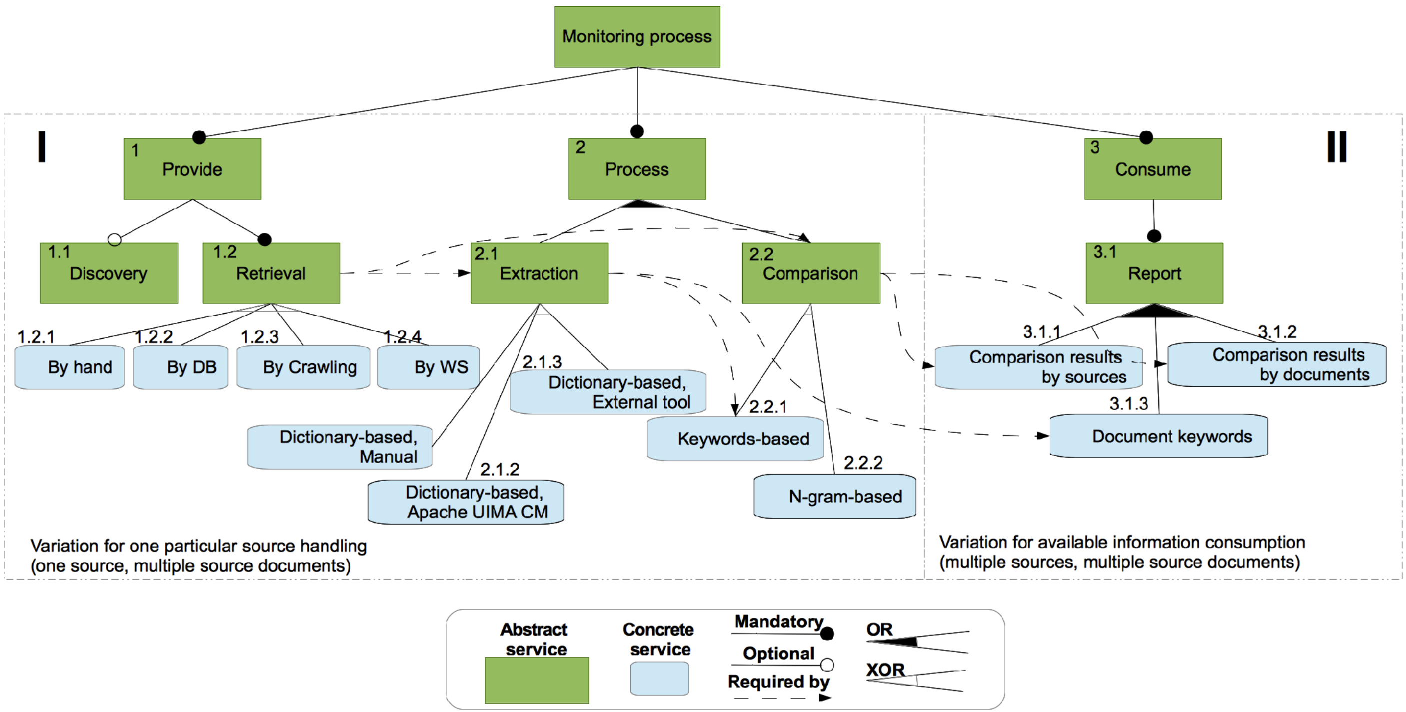

Fig. 2. Feature model for service system supporting monitoring process. Process is composed of three other information handling processes, namely, Provide, Process, and Consume. The features at the inner nodes represent groups of abstract services (variation points) realized by the concrete services (variants) at the leaves.

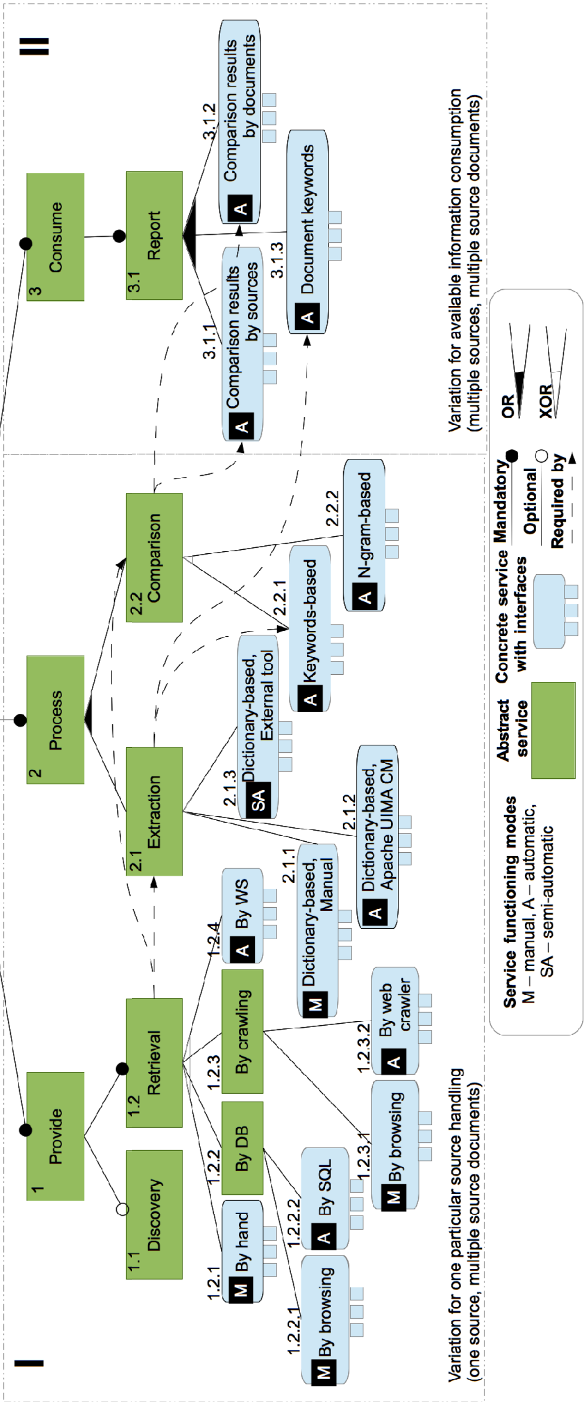

Fig. 3. Feature model for service system supporting monitoring process with the proposed FMA approach applied

As mentioned in previous section, service interfaces should also be analyzed for transition between services with the same or different functioning modes. In Table 2 we provide the variety of transitions available in EduMON system and illustrate particular modes of services in various compositions. Considering available functioning mode of the concrete service and its interfaces allows us to analyze the information service system more deeply to design or provide the input and output facilities or communication possibilities for passing and retrieving information to/from services with particular functioning mode. For instance, the service *Retrieval by hand* (see 1.2.1 in Fig. 3) with manual functioning mode is in transition with the service *Dictionary based extraction by Apache UIMA ConceptMapper* (see 2.1.2 in Fig. 3) with automatic functioning mode. 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, e.g., as input form where the human actor passes the document, e.g., study course description, to automatic keyword extraction service. In this particular example, the extended feature model (see Fig. 3) and possible transitions between services (see Table 2) helped to reason and take decision about the potential human involvement necessary to perform particular services.

Table 2. Available transitions between multi-mode services in EduMON. Numbers in table relate to numbers of services in Fig. 3. In the intersection of the row and column, the functioning modes of services in transition are indicated.

	1.2.1 (M)	1.2.2.1 (M)	1.2.2.2 (A)	1.2.3.1 (M)	1.2.3.2 (A)	1.2.4. (A)	2.1.1 (M)	2.1.2 (A)	2.1.3 (SA)	2.2.1 (A)	2.2.2 (A)	3.1.1 (A)	3.1.2 (A)	3.1.3 (A)
1.2.1 (M)							M-M	**M-A**	M-SA	M-A	M-A			
1.2.2.1 (M)							M-M	M-A	M-SA	M-A	M-A			
1.2.2.2 (A)							A-M	A-A	A-SA	A-A	A-A			
1.2.3.1 (M)							M-M	M-A	M-SA	M-A	M-A			
1.2.3.2 (A)							A-M	A-A	A-SA	A-A	A-A			
1.2.4. (A)							A-M	A-A	A-SA	A-A	A-A			
2.1.1 (M)												M-A	M-A	M-A
2.1.2 (A)												A-A	A-A	A-A
2.1.3 (SA)												SA-A	SA-A	SA-A
2.2.1 (A)												A-A	A-A	
2.2.2 (A)												A-A	A-A	
3.1.1 (A)														
3.1.2 (A)														
3.1.3 (A)														

5 Conclusions

In the paper we discussed how variability could be modeled in situations where functioning modes of services must be taken into consideration. Such situations arise in information service systems, which include services that can be performed manually, semi-automatically, and/or automatically. To facilitate design and implementation of these information service systems, we suggest to model their variability by feature model and propose FMA approach for extending the feature model by assigning functioning mode properties to the services. The approach uses concepts of abstract and concrete services and can dynamically convert variants corresponding to concrete services into variation points corresponding to abstract services.

The application of the approach is limited to the already designed variability model, i.e., it does not prescribe the creation of initial variability model. When the initial model exists, the FMA approach facilitates deeper analysis in design and implementation of information service systems by considering the degree of human involvement. In FMA approach we represented functioning modes of services as properties for the concrete services. Functioning modes can be captured also at the level of abstract services. It would require another way for feature model change management and would allow structuring the feature model differently with respect to functioning modes. The construction and examination of such feature model is one of our further research directions. Direct benefit of using extended feature model would be in multi-mode service composition where the feature model can be used to allow automatically deriving the multi-mode service flows permitted in the information service system. The implementation of multi-mode service composition based on the feature model is another research direction.

References

1. Galster, M., Weyns, D., Tofan, D., Michalik, B., Avgeriou, P.: Variability in Software Systems - A Systematic Literature Review. IEEE Trans. Softw. Eng. (2014)
2. Ralyté, J., Khadraoui, A., Léonard, M.: From information systems to information services systems: Designing the transformation. In: Grabis, J., Kirikova, M. (eds.) PoEM 2013. LNBIP, vol. 165, pp. 69–84. Springer, Heidelberg (2013)
3. Galster, M., Avgeriou, P., Tofan, D.: Constraints for the design of variability-intensive service-oriented reference architectures – An industrial case study. Inf. Softw. Technol. 55, 428–441 (2013)
4. Kang, K.C., Cohen, S.G., Hess, J.A., Novak, W.E., Peterson, A.S.: Feature-oriented domain analysis (FODA) feasibility study (1990)
5. Kang, K.C., Lee, H.: Variability Modeling. In: Capilla, R., Bosch, J., Kang, K.-C. (eds.) Systems and Software Variability Management: Concepts, Tools and Experiences, pp. 25–43. Springer, Heidelberg (2013)
6. Rudzajs, P.: Towards automated education demand-offer information monitoring: The system's architecture. In: Niedrite, L., Strazdina, R., Wangler, B. (eds.) BIR Workshops 2011. LNBIP, vol. 106, pp. 252–265. Springer, Heidelberg (2012)
7. Van Gurp, J., Bosch, J., Svahnberg, M.: On the notion of variability in software product lines. In: Proceedings of the Working IEEE/IFIP Conference on Software Architecture, pp. 45–54 (2001)
8. Svahnberg, M., van Gurp, J., Bosch, J.: A Taxonomy of Variability Realization Techniques: Research Articles. Softw. Pr. Exper. 35, 705–754 (2005)

9. Rudzajs, P., Kirikova, M., Strazdina, R.: Configurative Alignment of Business and Application Services: A Work Systems Perspective. In: Abramowicz, W. (ed.) BIS Workshops 2013. LNBIP, vol. 160, pp. 100–111. Springer, Heidelberg (2013)
10. Mohabbati, B., Asadi, M., Gašević, D., Hatala, M., Müller, H.A.: Combining service-orientation and software product line engineering: A systematic mapping study. Inf. Softw. Technol. 55, 1845–1859 (2013)
11. Lamprecht, A.-L., Naujokat, S., Schaefer, I.: Variability Management beyond Feature Models. Computer (Long. Beach. Calif). 46, 48–54 (2013)
12. Stollberg, M., Muth, M.: Service customization by variability modeling. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 425–434. Springer, Heidelberg (2010)
13. Petersen, K., Bramsiepe, N., Pohl, K.: Applying Variability Modeling Concepts to Support Decision Making for Service Composition. In: Service-Oriented Computing: Consequences for Engineering Requirements, SOCCER 2006, p. 1 (2006)
14. Nguyen, T., Colman, A.: A Feature-Oriented Approach for Web Service Customization. In: Proceedings of the 2010 IEEE International Conference on Web Services, ICWS 2010, pp. 393–400 (2010)
15. Khan, A., Kästner, C., Köppen, V., Saake, G.: Service variability patterns. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 130–140. Springer, Heidelberg (2011)
16. Khan, A., Kästner, C., Köppen, V., Saake, G.: Service Variability Patterns in SOC (2011)
17. Sasa, A., Juric, M.B., Krisper, M.: Service-Oriented Framework for Human Task Support and Automation. IEEE Trans. Ind. Informatics 4, 292–302 (2008)
18. Šaša, A., Krisper, M.: Service-oriented architectural framework for support and automation of collaboration tasks. J. Inf. Organ. Sci. 35, 119–133 (2011)
19. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man. Cybern. A Syst. Hum. 30, 286–297 (2000)