Approach for the Model Driven Development of Business Processes Lines Based on Service Oriented Architectures

Victor Cardona
MS. Student
Systems and Industrial Engineering Department
Universidad Nacional de Colombia
Bogotá, Colombia

Helga Duarte
PhD, Associate Professor
Systems and Industrial Engineering Department
Universidad Nacional de Colombia
Bogotá, Colombia

Abstract

Nowadays, organizations demand automation and flexibility of business processes for responding quickly to a great variety of emerging environmental changes. This need for flexible business processes demand systematic approaches to facilitate the implementation of business processes lines to cover effectively the growing needs of organizations. Two key factors in the design and implementation of agile business processes are reuse and productivity. This article describes a software process model called BplSoa (Business processes lines based on Service oriented architecture), which facilitates the development of business processes lines. The software process model mentioned describes systematically the steps required to implement business processes lines, emphasizing productivity through model-driven development and reuse through application of the principles of SPL (Software Product Lines).

Keywords: Software process model,business processes lines, model driven development, service oriented architecture.

1. Introduction

Competitiveness is a requirement for organizations seeking to meet the needs of their customers in a highly dynamic and globalized environment. The organizations have competitive advantages to cover the diversity of client needs when they implement continuous monitoring and improvement their business processes. The identification, description and automatization of business processes are essential for monitoring tasks performance and continuous improvement of business processes. The methodology of business process management – BPM, allows organizations to apply design techniques and improvement of business processes [Weske, 2007], which when is aligned with SOA (Service Oriented Architecture), the organizations obtain advantages translated in business agility [Haitham and Riad, 2008]. Currently software systems-oriented business processes, not only exist in one version that meets the global needs of a target market, however, there are many variations of software systems oriented processes, which specialize and differentiate by the divergent needs of customers [Schnieders and Puhlmann, 2006]. The study of the lines or families of business processes is a current area of research, which seeks to apply the principles of systematic reuse approach defined by SPL (Software Product Lines) in the development of high quality and low cost business processes.

Efforts have been made in research to contribute to the advancement of the study of the lines of business processes [Delgado, 2010; Flavio, Santana and Romero, 2009; Mohsen, 2009; Montero, 2008; Schnieders and Puhlmann, 2006], but currently, there is not a set of guidelines for guiding the analysis, design and implementation of business processes lines. This paper describes the research work that seeks to provide a software process model, to facilitate analysis, design and development of business processes lines supported on SOA, by applying the principle of high productivity of model-driven development and the mechanisms of systematic reuse defined by SPL.

Specifically the software process model is called BplSoa, which integrates under a unified and simple scheme, various contributions made by researchers in the area of business processes lines, for facilitating analysis, design and implementation of business processes lines.The paper is organized as follows: section 1.1 describes the motivation of research work, in section 2, we briefly review related concepts on which is based the present work, in section 3 is described the problem definition, the section 4 describes the software process model proposed, in section 5 is presented the related work and finally, section 6 presents some concluding remarks.

1.1 Motivation

The competitiveness of organizations is related to the effectiveness to provide timely solutions to customer needs, which depend largely on the maturity level of the organization to continuously improve their business processes. SOA plays a key role in the creation or modification of business processes in an organization, allowing reuse of existing business functionality for exposing it as a service. To carry out this transformation is necessaryto provide systematic mechanisms for service reuse [Flavio, Santana and Romero, 2009] to reduce costs and time of deployment of business processes based on SOA. There have been researches which seek to integrate the principles of software product lines (SPL) in the development of business processes. These works have provided advantages, such as increase of the quality and a minor time of development, covering design phase and implementation phase, besides briefly describing some technologies used in the implementation phases, giving as a result integration of SPL and business processes, but that do not cover the full cycle of software development [Boffoli 2009; Flavio, Santana and Romero, 2009; Mohsen, 2009; Schnieders and Puhlmann, 2006]. The above motivates the design of a software process model based on the principle of systematic reuse and high productivity, in order to facilitate the development of flexible business processes.

2. Basic concepts applied

2.1 Software Product Lines

A software product line (SPL) is a set of software-intensive systems sharing a common set of characteristics that satisfy the specific needs of a particular market segment and that are developed from a common set of core assets in a prescribed way [Clements and Northrop, 2002]. One strategy of SPL consists in systematic reuse of software artifacts to enhance product quality and decrease the development time of a new software product.

A product line is developed based on a predefined set of common software assets, highly reusable and customizable. The analysis, identification, designs and definition of a set of software assets requires an initial investment to achieve reuse and personalization in the development of specific products in a product line. The main motivations for creating a software product line are [Pohl, 2005]:

- Reduced development cost
- Reduced development time
- Increased quality
- Reduced maintenance effort
- Improve estimation of development effort

2.2 Business Processes Family Engineering

The business processes family engineering is a modern approach of software development that enables rapid development and deployment of cost-effective of process-oriented systems that fit the customer needs [Schnieders and Puhlmann, 2006]. The business processes family engineering, applies engineering techniques of software product lines to business process management, to develop business processes families. In this article we mention business processes lines to refer to business processes families or set of business processes that share some commonalities.

2.2 Model Driven Engineering

The goal of Model Driven Engineering – MDE, is effectively express the problem domain to combat the growing complexity of current software systems through [Douglas, 2006]:

- Domain specific modeling languages that allow formalize the application structure, behavior, and requirements within a particular domain.
- Transformation engines and generators that analyze certain aspects of the models and perform conversions of these models to lowest abstraction models for finally doing conversions from these models to source code.

3. Problem Definition

The objectives of the software industry are reducing costs, increasing quality, increasing productivity and reducing software development time. It has been shown that these objectives can be achieved with the application of principles of approaches as SPL (Software Product Lines) [Kentaro, 2010], and MDE (Model Driven Engineering) [Wim, 2003]. Therefore, the following research questions arise around business processes:

- How to integrate SPL as a mechanism for systematic reuse and MDE as mechanism to improve the productivity in a software process model to facilitate the development of business processes lines based on SOA?
- How to integrate existing tools to support the development of business processes lines based on SOA?

This article aims to contribute to the advancement of the study of the development of business processes lines, in particular, propose a software process model for facilitating the development of business processes lines based on SOA. The software process model integrates the principle of systematic reuse of SPL and the principle of high productivity of MDE. The software process model identifies and describes the main core assets and the activities required by the design and implementation of a line of business processes based on SOA.

4. BplSoa, Software Process Model Proposed

The software process model proposed is called BplSoa (Business processes lines based on Service oriented architecture), a model focused on model-driven development to facilitate the implementation of business processes lines. The software process model, BplSoa, is a software process centered on the principles of high productivity and reuse. High productivity is obtained by means of MDE through the use of models and model-to-model transformations and model-to-code transformations. Reuse is incorporated by systematic reuse principles defined through SPL. BplSoa guides and facilitates the development of business processes lines based on SOA, through clear identification and definition of engineering disciplines and work products involved in the software process. Table 1 describes the objectives, principles, benefits and disadvantages of the approaches MDE, SPL and SOA.

Table 1: Objectives, principles, benefits and disadvantages of MDE, SPL and SOA

Approach/ Characteristic	**MDE**	**SPL**	**SOA**
Objectives	-Combat the growing complexity of software systems platforms. [Douglas, 2006] -Express domain concepts effectively. [Douglas, 2006]	- Develop sets of products that together,satisfy the needs of a particular market segment or cover a particular mission, through the exploitation of the common features in various forms. [Clements and Northrop, 2002]	-Solve problems of integration and interoperability. [Papazoglou, 2006] -Increase organizational flexibility. [Carter, 2007]
Principles	-Increasing productivity in software development. [Douglas, 2006]	-Variability, customization and planned and systematic reuse of software assets. [Clements and Northrop, 2002]	Services principles [Erl, 2005]: -Service reusability -Service contract -Service loose coupling -Service abstraction -Service composability -Service autonomy -Service statelessness -Service discoverability
Benefits	-Increased productivity through model transformation engines and code generators. [Douglas, 2006] -Ease of development, debugging and development tools for the use of MDE. [Douglas, 2006]	-Increasing the quality of the products. [Clements and Northrop, 2002] -Reduced development time. [Clements and Northrop, 2002] -Reduced development costs. [Clements and Northrop, 2002] -Increased customer satisfaction. [Clements and Northrop, 2002]	-Improving the integration of heterogeneous systems. [Erl, 2005] -Promotes design reusable services. [Erl, 2005] -Leveraging the legacy investment. [Erl, 2005] -Organizational Agility. [Erl, 2005] -Establishment of a framework for vendor-neutral communication. [Erl, 2005]
Disadvantages	-Desynchronization of the models from code in later stages in a project. [Douglas, 2006]	- Cost associated with the design, implementation and maintenance of software assets. [Clements and Northrop, 2002]	-Lacks support for planned and systematic reuse [Flavio, Santana and Romero, 2009]

Based on the characteristics identified in Table 1, the following guidelines are identified to implement an integration strategy of the approaches MDE, SPL and SOA to facilitate the development of business processes lines:

- Apply mechanisms of planned and systematic reuse defined by SPL to design and implement services, in order to cover the lack of SOA to support planned and systematic reuse of services.
- Increase productivity in the implementation of business processes lines based on service oriented architecture, through the use of model-to-model and model-to-code transformations defined by MDE.
- Use feature models to define a set of business processes lines with common and variable characteristics that represent services with business logic functionality.
- Use BPMN (Business Process Model and Notation) [OMG BPMN 2011] models to represent business processes obtained through feature models.
- Use SoaML (Service Oriented Modeling Language) [OMG. SoaML 2012] models in order to represent service oriented architecture that supports the execution of business processes.

Figure 1 illustrates the overall strategy proposed in order to facilitate the design and implementation of a business processes line. The strategy proposes the use of feature models to represent the variability of the business processes line, the use of BPMN notation to model business processes, the use of SoaML to model the services architecture that supports the business processes, and proposes the use and implementation of transformations fm2bp (feature model to business process model based on BPMN notation), bp2soa (business process model to service oriented model), soa2code (service oriented model to services code).

In Figure 1 is illustrated the use of the approaches SPL, MDE and SOA for implementing business processes lines, as follows:

- SPL: Identification of common and variable characteristics of a line of business processes, where each feature represents an activity in a business process. Identification and modeling of the commons and variables characteristics is performed by means of a feature model that represents the set of business processes that make up the business processes line. Each variable and common characteristic in the feature model represents an activity that is subsequently mapped to a service within the SOA architecture required to support the execution of business processes.
- MDE: Using models, metamodels and transformations from model-to-model and from model-to-code, for improved productivity in development ofbusiness processes lines. The proposed metamodels for implementing business processes lines are: metamodel of feature model that is used to represent the business processes members of the line of processes. The resulting feature model is the input to obtain the business processes based on the BPMN metamodel. The business processes models in BPMN is the input of the transformation to a SOA model based on SoaMLmetamodel, which are used at the end of the process to generate the source code for each service of the solution, in order to support the execution of the business logic of each business process.
- SOA: mapping of each of the activities of the business process model to a service responsible for implementing the business logic. Through the SOA model are identified the providers, consumers of services, and the services architecture required to support the business processes.

The strategy described above is part of the software process model called BplSoa, with the aim of systematically identify the steps necessary to facilitate the implementation of business processes lines, through the application of the approaches SPL, MDE and SOA. Figure 2 illustrates BplSoa, dual software process model, which consists of the domain engineering cycle and application engineering cycle corresponding the two cycles in which is divided the product line engineering [Wim, 2003]. The following sections describe the engineering cycles of the software process model called BplSoa.

4.1 Domain engineering cycle

The main objective of the domain engineering cycle is the identification, design and implementation of reusable artifacts used in the application engineering cycle for derivation of business processes members of a business processes line. Artifacts may correspond to reference architecture, a model, an executable component, a deployment script, a test script or a requirements document [Kentaro, 2010].

The main reusable artifacts in BplSoa are the feature model that represents the business processes line and the model-to-model and model-to-code transformations.

4.1.1 Domain analysis discipline

The goal of domain analysis discipline is to identify, describe and modeling the set of business processes that compose the business processes line. The processes line is defined and represented by the variability model that is described next.

Variability model

The feature modeling is the most popular technique for domain analysis, which identifies and analyzes the commonalities and variabilities in a domain to develop highly reusable software assets for a software product line [Kwanwoo, 2002]. In the proposed software process model - BplSoa, the software products are the business processes, therefore, the modeling features for a business processes line is understood as representing its observable commons and variables characteristics. These observables characteristics in a business processes line are represented by tasks and sub processes of the business processes. Thus, the variability model is a feature model that represents all business processes of a business processes line in terms of the observable characteristics.

4.1.2 Domain design and implementation disciplines

The goal of domain design and implementation disciplines is to define the meta-models and transformations from model-to- model and from model-to-code required to support the implementation of business processes lines based on service oriented architecture. The transformations implemented become the main core assets needed to implement the members of the business processes lines in the application cycle of the software process model - BplSoa. The transformations are described next.

Feature model to business process model

In [Montero, 2008] was proposed an approach in order to map and obtain business processes models based on BPMN from feature models. The advantage of this transformation is to obtain directly a business processes model from a model that expresses the variability of a business processes line, but the proposal described in [Montero, 2008] for mapping between feature model and a business process model is insufficient because it does not allows to obtain a business processes model based on a simple sequence of activities. To cover the shortfall described above we propose include the sequence indicator as part of a feature name, in order to indicate the sequence or order of the given feature. The proposed sequence indicator is a number that is prepended to the feature name enclosed in square brackets. The sequence indicator is applied for all sub features of a given feature. Therefore, the sequence indicator is processed in order to map a feature model to a business processes model based on a sequence of activities. The Figure 3 illustrates the proposal for the mapping between feature models and business processes models based on the sequence indicator, with the purpose of supplementing the mapping proposed in [Montero, 2008].

Business process to service oriented model

The transformation from BPMN to SoaML is proposed in [Delgado, 2010], and its mapping strategy is presented in Table 2. Basically each activity on BPMN model is transformed to a service.

Table 2: BPMN to SoaML mapping proposed in [Delgado, 2010]

BPMN to SoaML mapping	
BPMN Element	***SoaML Element***
Business process model	SOA model
Pool or Lane	Participant
Activity	Service Point

Service oriented model to services code

The transformation from SoaML to code is done by tools that support SoaML meta-model like ModelPro™, MagicDraw™ Cameo SOA+ suite and Sparx Systems' Enterprise Architect [OMG. SoaML: Tool support]. The main advantage of the software process model called BplSoa is the use of standard meta-models in order to ensure compatibility and ease integration of tools of different vendors.

4.2 Application engineering cycle

In the application engineering cycle the main goal is the derivation of one or more members of the processes line defined in the domain engineering cycle. Core assets identified and implemented in the domain engineering are reused to speed up the derivation of business processes. The main assets reused by the application engineering cycle are the variability model, transformation of feature model to business process model (fm2bpm), transformation of business process model to SoaML model (bp2soa), and the transformation from SOA model to services code (soa2code).

4.2.1 Application analysis discipline

The goal of application analysis discipline is to choose the business process or processes to be implemented from the set of business processes represented by the variability model that was identified and defined in the domain engineering cycle through a feature model. The selection of the business process to be derived has to be aligned with the specific customer needs.

We introduce an initial case study for the admission of students to college. The case study consists of a business processes line that covers the admission process of different colleges. In the Figure 4 is illustrated the feature model of the selected process to be derived and its features are described next.

- Register: student could pay the admission fee and fill the form with personal and education information.
- Exam: the student present an exam of knowledge used to filter admission requests.
- Interview: the college could interview the aspirant.
- Evaluation: allows checking form, exam and the interview result in order to accept or reject the admission request.

4.2.2 Application design discipline

The goal of application design discipline is obtain the business process model and service oriented model needed by the implementation discipline in order to generate the code. The business process model illustrated in the Figure 5 is generated through the feature model presented in Figure 4 by means of the mapping strategy that was described in section 4.1.2. In the same way, the service oriented model illustrated in the Figure 6 is generated through the business process model presented in Figure 5 by means of the mapping strategy that was described in section 4.1.3. Each task in the BPMN model is mapped to a service, then, the model is detailed with the message types managed for each service.

4.2.3 Application implementation discipline

The goal of application implementation discipline is to generate the services code needed to support the execution of the business process that has been selected to be developed. The main inputs for this phase are the SOA model and the transformation soa2code (service oriented model to services code). There are a variety of tools that currently support the deployment of business processes like Intalio BPMS and WSO2 Business Process Server that facilitates the deployment, testing and execution of the business processes based on services and allow the execution of the business logic.

5. Related Work

The engineering approach to process families was proposed in [Schnieders and Puhlmann, 2006], and applies the product family engineering to business processes. The proposed approach introduces different variability mechanisms to business family or e-business families. The main difference between the proposal defined in [Schnieders and Puhlmann, 2006] and the present work is the management and representation of variability, while the approach described in [Schnieders and Puhlmann, 2006] manages the variability by means of the extension of the business process notation (BPMN), in the BplSoa model proposed,it is managed by means of the feature models, which are subsequently transformed (automatically)into business processes models based on BPMN.

The approach to development of SOA as Software Product Lines – SPL [Flavio, Santana and Romero, 2009], called service-oriented product lines, seeks to incorporate the principles of SPL to the development of SOA applications, to achieve high productivity, reducing development time of SOA applications customized to clients and specific market needs [Flavio, Santana and Romero, 2009]. One advantage of the model proposed in this paper with respect to the proposal described in [Flavio, Santana and Romero, 2009] is the approach towards the business processes as a strategy to generate value for companies, in terms of competitiveness through automation and flexibility of business processes.

Additionally, the software process model described in this paper focuses on MDE, in order to improve productivity through the application of transformations between models and between models and code.

The methodology for development of families of service oriented architectures proposed in [Mohsen, 2009], combines Software Product Lines – SPL with Model Driven Engineering – MDE, in order to cover the gap between business process management and software engineering. This work encourages the active use of model-to-model and model-to-code transformations in order to increase productivity during the derivation of members of the business processes lines.

6. Conclusions

The challenge for organizations lies in aligning goals and business strategies with technology. And a way to achieve this challenge is through automation, monitoring and constant optimization of their business processes. The study of families or lines of business processes seeks advantages such as high productivity, low cost and high quality in both, the analysis and the implementation of business processes lines. There has been progressandcontributions to the study of development of business processes lines, but still lacks integrated processes covering the full life cycle of the business processes lines. This article describes the software process model BplSoa (Business processes lines based on Service oriented architecture), proposed for facilitating the analyze, design and implementation of business processes lines based on the principles of service orientation, the principle of high productivity of MDE, by applying model-to-model and model-to-code transformations, and the mechanisms of systematic reuse of SPL. The main contribution of BplSoa is the intend to harmonize the principles provided by MDE, SPL and SOA approaches in a software process model in order to facilitate the implementation of agile business processes. Particularly, MDE gives to BplSoa the ability to separate the business and technology concerns [OMG SoaML 2012], so that, the models generated in the initial stages of design become reusable assets for the implementation of business processes in different technologies.

Additionally, this paper proposes a simple and novel approach to allow automatic transformation of feature models to business processes models, thereby contributing to the progress of the study of the business processes lines.

References

Carter, S. (2007). The New Language of Business: SOA & Web 2.0. IBM Press.

Clements P, Northrop L. (2002). Software Product Lines Practices and Patterns.Addison Wesley.

Delgado, A., García - Rodríguez de Guzmán, I., Ruiz, F., Piattini, M. (2010). "Tool support for Service Oriented development from Business Processes", 2nd International Workshop on Model-Driven Service Engineering (MOSE’10), Málaga, Spain.

Douglas, C. Schmidt. (2006). Model-driven engineering. In IEEE Computer Society, pages 25 – 31.

Erl, Thomas. (2005). Service-Oriented Architecture: Concepts, Technology, and Design. UpperSaddleRiver, NJ, USA: Prentice Hall.

Flavio Mota Medeiros, Eduardo Santana de Almeida, Silvio Romero de Lemos Meira. (2009). Towards an Approach forService-Oriented Product Line Architectures.13th International Software Product Line Conference (SPLC).

Haitham Abdel Monem El-Ghareeb, A. M. Riad. (2008). Aligning Service Oriented Architecture and Business ProcessManagement Systems to Achieve Business Agility. Egyptian informatics Journal, Cairo University, Vol. 9, Issue 1.

Kentaro Yoshimura Paul Jensen John D. McGregor, Dirk Muthig. (2010). Successful software product line practices. IEEE SOFTWARE, 27:16–21.

Kwanwoo Lee, Kyo C. Kang, and Jaejoon Lee. (2002). "Concepts and Guidelines of Feature Modeling for Product LineSoftware Engineering", in Proc. ICSR, pp.62-77.

Mohsen Asadi, BardiaMohabbati, NimaKaviani, DraganGasevic, Marko Boskovic, MarekHatala. (2009).

Model-DrivenDevelopment of Families of Service-Oriented Architectures. ACM International Conference Proceeding Series archive Proceedings of the First International Workshop on Feature-Oriented Software Development. Denver Colorado.

Montero I, Pena J, Ruiz-Cortes A. (2008). "From Feature Models to Business Processes," in Proceedings of the IEEEInternational Conference on Services Computing Vol. 2.pp.605-608.

Nicola Boffoli, Marta Cimitile, Fabrizio Maria Maggi, Giuseppe Visaggio. (2009). Managing SOA System Variation throughBusiness Process Lines and Process Oriented Development. 13th International Software Product Line Conference (SPLC). August 24–28. San Francisco, CA, USA.

OMG. BPMN: Business Process Model and Notation, v2.0, (2011). http://www.omg.org/spec/BPMN/2.0/

OMG.SoaML: Service oriented architecture Modeling Language. v1.0, (2012).http://www.omg.org/spec/SoaML/1.0/PDF

OMG.SoaML: Tool support. (2010).http://www.omgwiki.org/SoaML/doku.php?id=tool_support

Papazoglou, M. Willem-Jan van den Heuvel. (2006). "Service-oriented design and development methodology," InternationalJournal of Web Engineering and Technology (IJWET), vol. 2, no. 4, pp. 412–442.

Pohl Klaus, van der Linden Frank. (2005). Software Product Line Engineering: Foundations, Principles and Techniques.Springer-Verlag, New York, Inc, USA.

Schnieders A, and Puhlmann F, (2006). "Variability mechanisms in e-business process families," 9th International Conference on Business Information Systems.

Weske, M. (2007). Business Process Management: Concepts, Languages, Architectures. Springer-Verlag Berlin Heidelberg.

WimBast, AnnekeKleppe, Jos Warmer. (2003). MDA Explained, The Model Driven Architecture: Practice and Promise.Addison-Wesley.

Figure 1: Integration strategy of approaches MDE, SPL, and SOA in the software process model proposed called BplSoa.

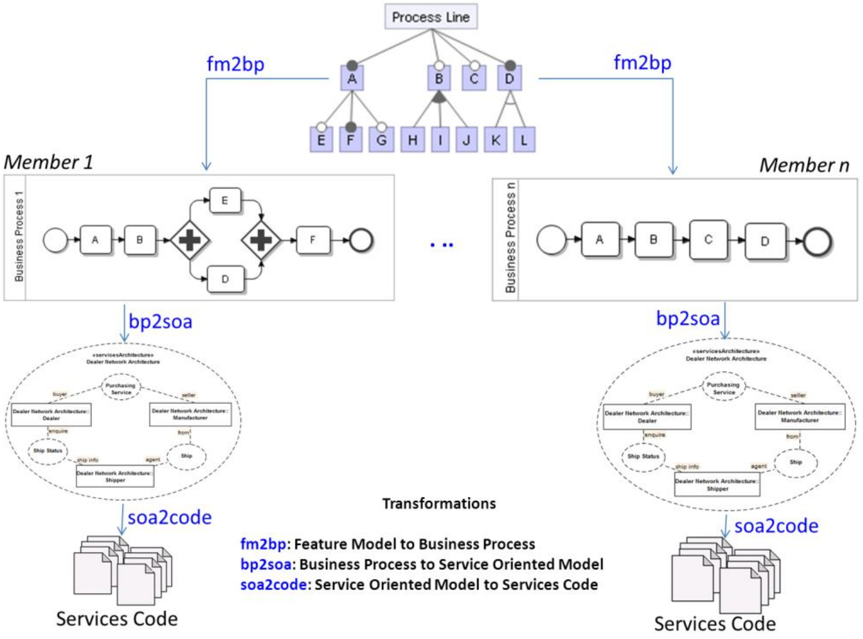

Process Line
fm2bp
fm2bp
A
B
C
D
E
F
G
H
I
J
K
L
Member 1
Member n
Business Process 1
Business Process n
A
B
E
D
F
C
. . .
bp2soa
bp2soa
«servicesArchitecture»
Dealer Network Architecture
Purchasing Service
buyer
seller
Dealer Network Architecture::Dealer
Dealer Network Architecture::Manufacturer
enquire
from
Ship Status
Ship
ship info
agent
Dealer Network Architecture::Shipper
soa2code
soa2code
Services Code
Services Code
Transformations
fm2bp: Feature Model to Business Process
bp2soa: Business Process to Service Oriented Model
soa2code: Service Oriented Model to Services Code

Figure 2: Software process model proposed called BplSoa

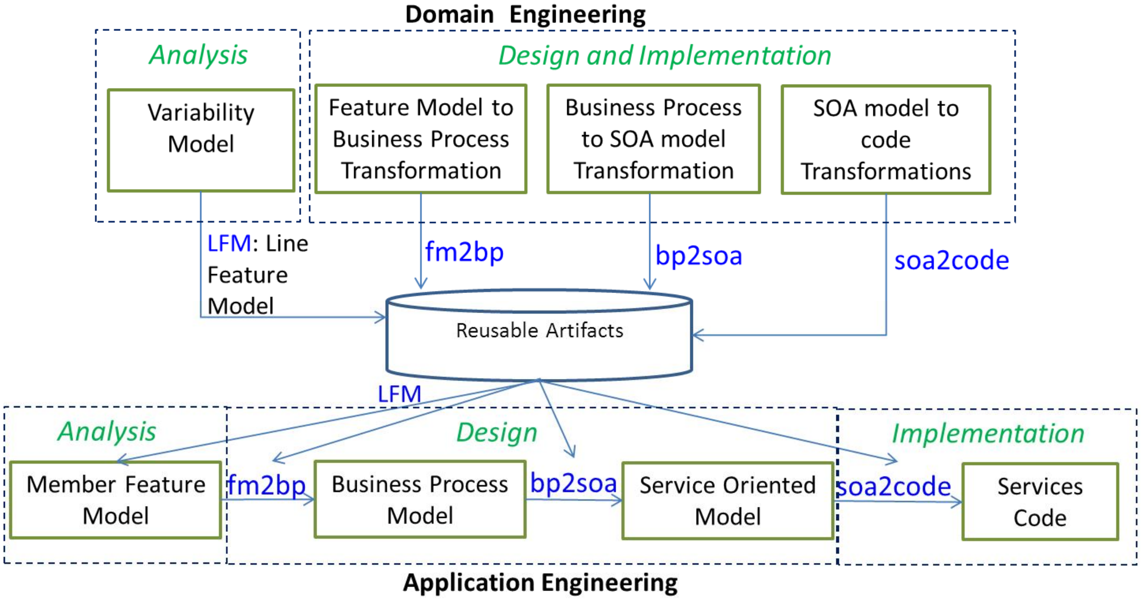


Figure 3: Feature model to business process model mapping

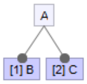
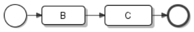
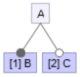
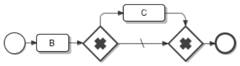
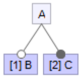
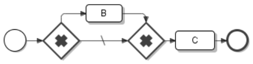
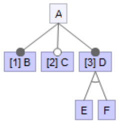
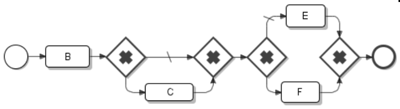
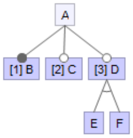
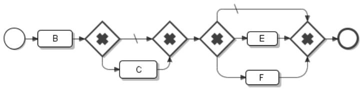
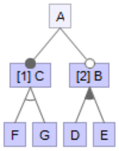
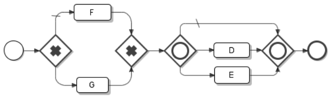
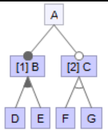
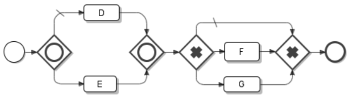
Feature Model	Business Process Model
 Mandatory relation	 Sequence of task B and C.
 Mandatory and optional relation	 Task B and optional task C
 Optional and mandatory relation	 Optional task B and task C
 Mandatory, optional and alternative-mandatory relations	 Task B, and optional task C and mandatory exclusive tasks E - F
 Mandatory, optional and alternative-optional relations	 Task B, and optional task C, and optional exclusive tasks E - F
 Mandatory-alternative and or-optional relations	 Mandatory exclusive tasks F-G, and optional inclusive tasks D-E
 Or-mandatory and optional-alternative relations	 Mandatory inclusive tasks D-E, and optional exclusive tasks F-G

Figure 4: Feature model of student admission process

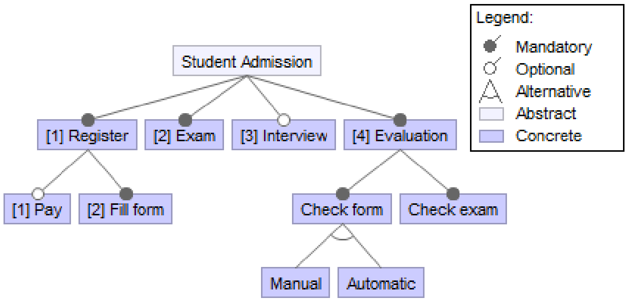


Figure 5: Business process model of student admission

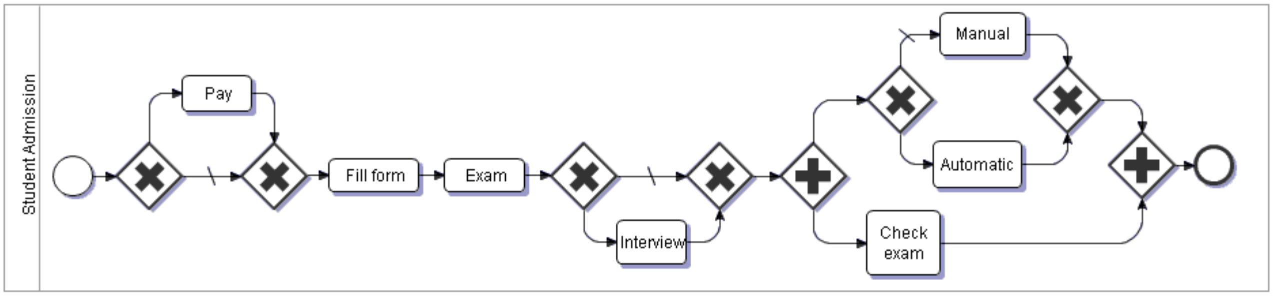
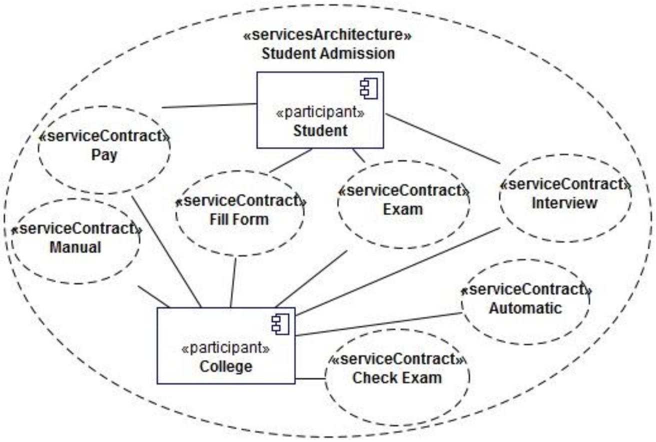


Figure 6: Service oriented model of student admission


«servicesArchitecture»
Student Admission
«participant»
Student
«serviceContract»
Pay
«serviceContract»
Fill Form
«serviceContract»
Exam
«serviceContract»
Interview
«serviceContract»
Manual
«serviceContract»
Automatic
«participant»
College
«serviceContract»
Check Exam