Managing SOA System Variation through Business Process Lines and Process Oriented Development

Nicola Boffoli, Marta Cimitile, Fabrizio Maria Maggi, Giuseppe Visaggio

Department of Informatics - University of Bari - Via E. Orabona 4 - 70126 - Bari - Italy

(boffoli, cimitile, maggi, visaggio)@di.uniba.it

Abstract

Software Product Lines (SPL) and Service-Oriented Architectures (SOA) are two emerging approaches to the software development currently receiving great attention both in research and in practice.

Our work suggests an approach to transfer the main peculiarities of the SPL (i.e. asset reuse and variation mechanisms) to the SOA systems development, in order to realize a SOA systems line. In this way we provide a method to easily adapt a SOA application to different customer needs in changeable environments.

All this is realized using the Business Process Lines (BPL) concept together with the Process Oriented Development (POD) paradigm. A BPL realizes process models suitable to different customers or market segments needs. The POD paradigm allows to transform a process model into a SOA system.

Finally we show an application of our proposal in a research project that involve several industrial and academic organization. In the project a set of BPL is realized and implemented using the MIT process handbook.

1. Introduction

Software Product Lines (SPL) [1] and Service Oriented Architectures (SOA) [2] aim to develop software systems through two common perspectives: software reuse and flexibility [3]. Using these approaches, enterprises can implement software systems for different customers reusing software resources rather than developing the same software capabilities again. In this way they gain in productivity, software quality and time to market.

A SPL is a set of software-intensive systems sharing common features. In particular a SPL aims to satisfy the specific needs of a market segment using a common set of core assets in a prescribed way.

In general, the SPL paradigm is characterized by two different concepts:

- **Asset Reuse**: management of the “Core Asset” i.e. collection, organization and systematic refinement of the invariant or variant assets representing respectively the SPL Commonality and Variability.

- **Variation Mechanisms**: automatic building of the products based on the systematic reuse of the “Core Assets”. Each asset is a software component with fixed specifications allowing to:
  - **Configure** the products through asset integration procedures;
  - **Specialize** the assets through the specification of their parametric parts.

A SOA is a software architecture able to orchestrate web services to guarantee the integration of heterogeneous systems in a business process. A SOA is made up of components and interconnections stressing interoperability and location transparency.

Early research works, concerning the comparison between SPL and SOA, are just appearing in the software engineering research community [4]. In particular a crucial research question is: how can SOA systems benefit from SPL good practices (i.e. reuse and variation management approaches)? Or in other words what is the core asset and how can we implement the variation mechanisms in the SOA system context?

![Figure 1: SPL concepts in SOA context](image)

In this paper we answer these questions through the approach illustrated in Figure 2. In particular, we propose transferring advantages from SPL to SOA operating on business processes and using two specific instruments:

1. **Business Process Line (BPL)**. A BPL according to the SPL practices is able to model an appropriate business process, process variant, suitable for specific customer needs.
2. **Process Oriented Development (POD)**. POD is able to transform a process variant into an executable SOA system through successive transformations aimed at making the process model understandable by an execution engine.

The resulting SOA system line automates the underlying business processes; so, if we adapt the business processes to new customer needs using the underlying BPL and then we generate from it the corresponding SOA system, it will result in its turn suitable to the specific customer requirements. Moreover since the SOA system results from the translation of the former process models, it will be in compliance with the underlying business processes without misalignments between models and their implementation.

The remainder of the paper is structured as follows: section 2 presents the overview of the proposal and discusses in detail the BPL and POD approaches; in section 3 is described the application of the approach in a research project; section 4 completes the paper providing some conclusive insights and final remarks showing prospective works.

### 2. Proposed Approach

The proposed approach is synthesized in Figure 2. It consists in two main phases using respectively the Business Process Lines concept and the Process Oriented Development paradigm.

In the first phase, starting from the current customer needs, a BPL allows to realize a process variant specific for the given requirements. In the second phase, starting from this model, POD allows to automate this model and transforms it into a SOA system. These phases are detailed in the following paragraphs.

#### 2.1 Business Process Lines (BPL)

In [5, 6, 7] the authors transfer the main SPL concept to the business processes field.

According to these works a BPL could be considered a set of similar business processes sharing a common part (commonality) and characterized by a variant part (variability) depending on the specific context where the process will be applied (more details in [8]). In particular a BPL works integrating a set of process assets, i.e. atomic reusable parts of a business process (one or more activities with their IN/OUT). Commonality is a set of invariant assets and variability is a set of variant assets selected according to a fixed context profile.

Commonality and variability are then integrated in order to obtain a process variant to be applied in the given context. The assets integration rules are driven by their IN/OUT artifacts allowing to establish the succession of the process assets: the outputs of the previous asset are the inputs of the successive one.

The details of this phase are represented in Figure 3.

When a BPL is selected the invariant assets and all the candidate variant assets are specified. The BPL is selected on the basis of the customer specific needs: for example if we are interested to the process “Selling” we will use a “Selling BPL”. The invariant assets of the “Selling BPL” could be for example “Obtain Order”, “Deliver product” and “Receive Payment”. Each of these process assets is composed by the basic activities, inputs and outputs necessary within the “Selling” process. Afterwards to identify a specific process variant, the variant assets have to be selected among the candidate ones on the basis of a specific context. For example if we want to sell via
2.1.1 BPL Logical Model

To choose the suitable process variant of a BPL to be applied in a specific context profile, we need a function associating to a specific context profile the process variant specific for the given context:

$$f: CP \rightarrow S$$

where

- $CP$ is the set of all the possible context profiles. An element $cp \in CP$ is represented as a vector of instantiated diversity factors $DF_i = [df_{i1}, df_{i2}, \ldots, df_{iq}]$ where each $df_{ij} = 1, \ldots, q$ is an instance of $DF_i$. So we can say that the set $CP$ is:

$$CP = [DF_1] \times [DF_2] \times \ldots \times [DF_r].$$

- $S$ is the set of all the possible process variants of the considered BPL.

$f$ can be detailed in this way:

$$f(cp) = \Phi(\sigma(cp, K), \sigma(cp, \chi(cp)))$$

If $A$ is the set of all the process assets associated to the BPL, in (2):

- $K = \{ia_1, ia_2, \ldots, ia_n\} \subseteq A$ is the set of invariant assets realizing the commonality;

- $\chi$: $CP \rightarrow CVA = A-K$ is the function referred to the activity 1.2 in Figure 3 associating to each fixed $cp \in CP$ the set of variant assets $\{va_1, va_2, \ldots, va_m\} \subseteq CVA$ realizing the process variability according to the fixed context $cp$.

$CVA = A-K$ is the set of the candidate variant assets associated to the BPL;

- $\sigma$ is the function referred to the activity 1.3 in Figure 3 including the assets specialization rules on the basis of the context profile $cp$. It associates to a set of assets another set of assets specialized according to the fixed context profile. In particular $\sigma(cp, \{asset_1, asset_2, \ldots, asset_p\}) = \{\sigma_1(cp, asset_1), \sigma_2(cp, asset_2), \ldots, \sigma_p(cp, asset_p)\}$, where $\sigma_1, \sigma_2, \ldots, \sigma_p$ are transformations specializing respectively the assets $asset_1, asset_2, \ldots, asset_p$ according to the context profile $cp$.

- $\Phi$ includes the integration rules useful to compose the commonality to the variability specialized according to the context profile $cp$.

So, $f(cp) = \Phi(\sigma(cp, K), \sigma(cp, \chi(cp))) = \Phi(\sigma(cp, \{ia_1, ia_2, \ldots, ia_n\}), \sigma(cp, \{va_1, va_2, \ldots, va_m\})) = \Phi(\{\sigma_1(cp, ia_1), \sigma_2(cp, ia_2), \ldots, \sigma_p(cp, ia_n)\}, \{\sigma_1(cp, va_1), \ldots, \sigma_m(cp, va_m)\})$ identifies the process variant suitable for the context profile $cp$.

2.1.2 BPL Operative Model

The functions $\chi$ and $\sigma$ defined in the logical model could be implemented through a decision tables system (more details in [9, 10, 11, 12]).

A Variability Selection table implements the function $\chi$ of the logical model. It allows to select the suitable variant assets composing the variability, characteristic of a specified context profile. So this kind of decision table is structured as follows (Figure 4):

- the CONDITION quadrant contains the diversity factors $DF_i, i = 1, \ldots, r$ driving the variant assets selection;

- the CONDITIONAL STATE quadrant contains the possible value of each diversity factor: $[DF_i] = \{df_{i1}, df_{i2}, \ldots, df_{iq}\}$;

- the ACTION quadrant contains all the candidate variant assets ($CVA$) that can be selected to realize the process variability;

- the RULE quadrant identifies the relationships between each context profile and the variant assets to realize the corresponding process variability.

![Figure 4: Variability Selection table](image-url)
An Asset Specialization table implements a function \( \sigma_i \) of the logical model. It allows to specialize a process asset (variant or invariant) on the basis of specified context profile, executing a set of specializing actions. So this kind of decision table is structured as follows (Figure 5):

- the CONDITION quadrant contains the diversity factors \( DF_i, i=1,\ldots,r \) driving the asset specialization;
- the CONDITIONAL STATE quadrant contains the possible values of each diversity factor: \( DF_i = \{df_{i1}, df_{i2}, \ldots, df_{ir}\} \);
- the ACTION quadrant contains the actions to specialize the asset according to the specified context profile;
- the RULE quadrant identifies the relationships between each context profile and the specializing actions to be applied.

### Figure 5: Asset Specialization table

<table>
<thead>
<tr>
<th>( DF_1 )</th>
<th>( df_{11} )</th>
<th>( df_{12} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( DF_2 )</td>
<td>( df_{21} )</td>
<td>( df_{22} )</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( DF_n )</td>
<td>( df_{n1} )</td>
<td>( df_{n2} )</td>
</tr>
</tbody>
</table>

---

\( sa_1 \)  
- x  
- x  
- x  
- x

---

\( sa_m \)  
- x  
- x  
- x  
- x

---

\( sa_n \)  
- x  
- x  
- x  
- x

2.2 Process Oriented Development (POD)

Starting from the process variant identified through a BPL, a SOA system can be implemented using the POD paradigm. In particular the process model is made understandable by an execution engine through successive transformations. In this way the POD allows to implement in a SOA system the changeable process requirements captured using a BPL, so a SOA systems line is actually realized. Referring to Figure 2, POD implements a SOA system starting from the process variant that is the output of the BPL.

Figure 6 illustrates in details the sequence of activities to realize this implementation. The abstract process model, formally described through a Process Modeling Language, is enriched with implementative details realizing a new intermediate model called Detailed Process Model (DPM). Finally a BPEL workflow is created as the last and most specific model.

Referring to Figure 6, in the Specification activity the process variant is translated into the corresponding DPM. As described above, the DPM will be realized enriching the process variant through implementative details making it understandable by an automatic translator. In particular a process model describes how activities are carried out exchanging artifacts. So in the DPM each artifact is mapped with a BPEL variable. Each variable has a specific type. This variable type should be traceable with the characteristic of the artifact. The information about an artifact in a process model are not enough to translate the artifact in a BPEL variable, that’s why this association requires the pre-specification defined in the DPM. Moreover the DPM contains the mapping between the activities of the process model and the different services implementing them.

![Figure 6: Process Oriented Development Approach](image)

We want to highlight that real processes could provide the occurrence of human activities as well. So to implement the process activities executed by human agent, an extension of BPEL, BPEL4People, is used. In particular in the DPM the URL of the services carrying out the automatic activities and the e-mails of the agents carrying out the human activities should be specified. In this way the BPEL workflow will invoke the suitable software application to execute the automatic activities and will inform by e-mail the suitable agent to execute the human activities. The Web Services Specification produced by the Specification phase is useful to identify the services implementing the automatic activities (Web Services Selection/Development).

Starting from the information specified in the DPM the translator is able to generate an executable BPEL workflow (Workflow Development) implementing the underlying business process using the specified web services (Integration). For reasons of brevity we don’t explain in details the translation algorithm but we can
sketch the mapping between the elements of a business process and the corresponding BPEL objects. Table 1 shows the mapping between the process elements and the BPEL objects translating them. In the workflow BPEL the connectors linking all these elements are translated using the BPEL tag <link>.

<table>
<thead>
<tr>
<th>Process Element</th>
<th>BPEL object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Node</td>
<td>&lt;Receive&gt;</td>
</tr>
<tr>
<td>End Node</td>
<td>&lt;Reply&gt;</td>
</tr>
<tr>
<td>Activity</td>
<td>&lt;Invoke&gt;</td>
</tr>
<tr>
<td>Decision Node</td>
<td>It could be translated as a tag BPEL &lt;if&gt;, &lt;while&gt; or &lt;repeat until&gt; on the basis of the specific control flow</td>
</tr>
</tbody>
</table>

In particular an activity is translated using the BPEL tag <Invoke>. In a process model activities could be atomic or composite. The process model could be seen as a tree where atomic activities are the leaf-nodes. The translation algorithm starting from the leaf-nodes of the tree cover recursively the whole process model. Each atomic activity corresponds to a web service invocation. Each composite activity corresponds to the invocation of a BPEL workflow orchestrating the web services implementing its sub-activities that could be atomic or in their turn composite.

3. Case Study: a Selling SOA systems line

The proposed approach has been adopted in a research project. In Puglia some industrial and research organizations are working on it. They are collaborating to implement SOA systems so that business processes of different customers in the local agricultural and food market can be automated. Within this project there are different providers offering a number of software services that could be reused and adapted in each different business contexts according to the customer needs. That’s why the project represents a field of interest for the proposed approach application.

The application of the proposed approach in the project has been realized through three main steps:

- BPL definition
- BPL application
- POD application

The BPL definition step consists in the extraction of a set of BPL aiming to model different kinds of business processes.

The BPL application consists to apply, starting from the defined BPL, the approach proposed in Figure 3. Finally the POD application consists in the application of the approach proposed in Figure 6 to the process variant obtained in the BPL application step.

3.1 BPL Definition

Starting from the analysis of MIT library [13] we have achieved the information useful to obtain a BPL Library (we will deepen these outcomes in a future work). The MIT library represents a collection of more than 5000 business activities related in several business processes. These activities and processes have been used to obtain a set of BPL allowing to model several kinds of business processes.

In particular we describe in this session, the definition of a Selling BPL obtained using the MIT Selling processes.

<table>
<thead>
<tr>
<th>Invariant assets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ia&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td>ia&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td>ia&lt;sub&gt;3&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

Starting from MIT Selling processes we have identified, through their common parts, the invariant assets realizing the Selling BPL commonality (Table 2). Each process asset is represented as one or more activities and their IN/OUT: for instance the asset ia<sub>1</sub>=“Obtain Order” is represented as in Figure 7.

![Figure 7: “Obtain Order” asset](image)

Afterwards using the information about the different application contexts where the MIT processes are applied we have been able to define a number of diversity factors and their possible values (Table 3) to specify the conditions of the variability selection table.

<table>
<thead>
<tr>
<th>Diversity Factors</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sell How</td>
<td>Sell via physical store</td>
</tr>
<tr>
<td></td>
<td>Sell via electronic store</td>
</tr>
<tr>
<td></td>
<td>Sell via face-to-face</td>
</tr>
<tr>
<td></td>
<td>Sell via direct mail</td>
</tr>
<tr>
<td></td>
<td>Sell via email/fax</td>
</tr>
<tr>
<td></td>
<td>Sell via television direct response marketing</td>
</tr>
<tr>
<td></td>
<td>Sell via telemarketing</td>
</tr>
<tr>
<td>Sell What</td>
<td>Service, Product, Process</td>
</tr>
<tr>
<td>Auction</td>
<td>Y, N</td>
</tr>
<tr>
<td>Advance Payment</td>
<td>Y, N</td>
</tr>
<tr>
<td>Quality control</td>
<td>Y, N</td>
</tr>
<tr>
<td>Selling Suggestions</td>
<td>Y, N</td>
</tr>
</tbody>
</table>
Finally using the variable parts of the MIT Selling processes, we have identified the candidate variant assets (Table 4) to specify the actions of the variability selection table. Using this information, we have been able to realize the variability selection table (Figure 8). It encloses the rules to associate to each possible context profile the related variant assets. These have to be composed with the BPL variant assets to obtain the process variant specific for the specified context profile.

Table 4: Candidate variant assets

<table>
<thead>
<tr>
<th>vb1</th>
<th>Share out goods</th>
</tr>
</thead>
<tbody>
<tr>
<td>vb2</td>
<td>Register Seller</td>
</tr>
<tr>
<td>vb3</td>
<td>Register Alternative Products</td>
</tr>
<tr>
<td>vb4</td>
<td>Arrange store displays</td>
</tr>
<tr>
<td>vb5</td>
<td>Auction</td>
</tr>
<tr>
<td>vb6</td>
<td>Check quality</td>
</tr>
<tr>
<td>vb7</td>
<td>Register Auction Result</td>
</tr>
<tr>
<td>vb8</td>
<td>Identify potential customer needs</td>
</tr>
<tr>
<td>vb9</td>
<td>Identify potential customers</td>
</tr>
<tr>
<td>vb10</td>
<td>Inform potential customers</td>
</tr>
<tr>
<td>vb11</td>
<td>Manage customer relationships</td>
</tr>
</tbody>
</table>

For instance, considering the column 5 of the table, we have that for the context profile $cp^* = (Sell via physical store, Product, Y, N, Y, Y)$ the variant assets to select are: $va_1 =$ "Share out goods", $va_2 =$ "Register Seller", $va_3 =$ "Suggest Alternative Products", $va_4 =$ "Arrange store displays", $va_5 =$ "Auction", $va_6 =$ "Check Quality", $va_7 =$ "Register Auction Result", $va_8 =$ "Identify potential customer needs", $va_9 =$ "Identify potential customers", $va_{10} =$ "Inform potential customers". So to obtain the process variant specific for the given context profile we have to compose these assets with the invariant assets of the commonality.
Moreover to define the asset specialization tables for each process asset we have identified the specializing actions corresponding to each possible context profile. Since the conditions are the same for each process asset, we can incorporate all the asset specialization tables corresponding to the functions $\sigma_i$ in only one table (corresponding to the function $\sigma$ of the logical model).

The resulting table (Figure 9) encloses the rules to associate to each context profile, the actions to be executed to specialize the behavior of all the assets related to the considered BPL.

For instance, according to the context profile $cp^*$ considered before (column 5), we need to specialize the activities: “Deliver” in “Deliver product”, “Receive Payment” in “Receive Payment at register”. Moreover we have to specialize the artifact “Advertising initiatives” in “Physical Store advertising”. Finally we have to add the input “Auction Sticker” and “Shipping paper” respectively to “Receive Payment” and “Deliver Product activities”.

### 3.2 BPL Application

The obtained BPL could be used to model different process variants of the Selling process. During the research project this BPL has been used to model different process variants to fulfill different customers involved in the project. In particular, in this work we describe the application of the Selling BPL to automate the Selling process of a fish consortium in Puglia. The consortium was interested to sell via auction the fish caught by the ship-owner members boats. In particular the organization needed to model its Selling process and automate it in a SOA system integrating the services of different providers.

In this scenario the BPL has allowed to model the customer Selling process easily. According to the first activity of Figure 3, for the customer was necessary to model a Selling process. In this case we have been able to use the BPL defined in the paragraph 3.1. For this BPL, the invariant assets are listed in table 1 and the candidate variant assets are listed in table 3.

Afterwards the variability selection table has been executed. Customer requirements have allowed to specify the context profile $cp^*$ corresponding to column 5 in Figure 6. In this way we have obtained the corresponding list of variant assets.

Finally the assets specialization activity has been performed using the Specialization Table shown in Figure 7. We have identified the list of specialization actions (see column 5 of the Specialization table). So each asset has been specialized through a set of specialization actions. Finally, all the specialized assets have been integrated considering their IN/OUT.

The obtained process model is shown in Figure 10.

### 3.3 POD Application

According to the POD paradigm (Figure 6) we have realized the DPM starting from the process variant identified through the BPL. This new model has been obtained enriching the process model with implementative details. For this purpose we have implemented an application, ExportBPEL to support the DPM generation. This application has been realized as add-in of Enterprise Architect (EA), a graphical UML design and business analysis tool for modeling, documenting, building and maintaining object-oriented software systems.

In Figure 11 a screenshot of the application is shown. Here the implementation details related to the activity 7. Register_Auction_Result are defined. In particular the developer has to specify the variable types implementing the input and output artifacts of the activity and the method implementing the activity itself. In this way all the specifications to realize the BPEL workflow are
identified. In this case the activity must be implemented through the method `method_7_Register_Auction_Result` providing two input variables: `Entry_Document_ID` (String) and `Auction_Winner_Username` (String). The output variable must be a complex type composed by `Auction_Sticker_ID` (String) and `Shipping_Paper_ID` (String).

On the other hand Software Specification to realize the web services implementing the different activity have been realized. The different services provider have used these specifications to develop the suitable web services. Starting from the DPM previously created ExportBPEL allows to realize automatically the Workflow BPEL implementing the underlying business process. Finally the BPEL workflow has been completed specifying the URL of the web services implemented by the different providers.

![Figure11: Export BPEL application](image)

4. Conclusions

This work represents a contribution to transfer the good practices of SPL (asset reuse and variation mechanisms) to SOA system development. The adoption of BPL and POD approaches permits to select the suitable process variant and to implement the appropriate SOA system for one or more customers.

In particular in this work we refer to the automation of the Selling process in a fish consortium. In this scenario we have evaluated some advantages deriving from the adoption of the proposed approach:

- it facilitates the selection of the suitable process model according to the customer requirements reducing its time and effort of 80%;
- it facilitates the implementation of the selected business process reducing its time and effort of 30%.

These values are qualitative evaluations on the basis of the feedbacks of the experts involved in the project. They compare time and effort for the modeling and implementation steps using our approach with the corresponding data obtained in similar cases performed with traditional approaches.

The described research project is not yet concluded. BPL and POD will apply in other business processes in the same research project to confirm and deepen these results.

References