

# Situational Method Engineering applied to Business Modelling

José Semedo  
jose.semedo@ist.utl.pt  
INESC INOV  
Rua Alves Redol, 9 - 1000-029 Lisboa – Portugal

**Abstract.** The business modelling of an enterprise involves multiple business stakeholders and their respective concerns. To model all the concepts addressed to an organisation's business we use Business Process Modelling Languages in order to obtain different views over the same Business Process Diagram, or the same Enterprise Architecture Diagram. The views are obtained from viewpoints based on the stakeholders' concerns. In this project we aim to use the Situational Method Engineering to conceptualize a library with chunks of viewpoints, representing steps of methods to obtain views over business process diagrams on a multi-stakeholder perspective.

**Keywords:** Situation, Viewpoint, View, Representation, Multi-stakeholder, Concern, Business Process Modelling

## 1 Introduction

An Enterprise Architecture of a System is described by an Enterprise Description. That description has models and identifies multiples Stakeholders – System Stakeholders. Each stakeholder has their concern or concerns leading to a set of Viewpoints, represented by Views. Therefore an Enterprise Description is organised and described by a set of Views of System models. Taking those concepts to the Business layer of the Enterprise Architecture, we will face multiples stakeholders and their respective concerns which lead to a set of visualizations over one Business Process Model (BPM) or multiple ones. We call the set of visualizations or representations as views [1].

The engineering of information systems development is different in each situation. Information Systems Development (ISD) depends of different organisational, technical and human factors, so the complexity of each varies. Therefore, there is no universal and unique method to perform information systems engineering, we need to use, adapt and configure specific methods and tools for each situation. In this perspective we have the Situational Method Engineering (SME), which is a methodology that focuses on the creation of the specific methods on the fly according to the ISD situation, instead of looking for universally applicable ones. There are some approaches of SME but most of them promote de construction and adaptation of new methods by assembling reusable method fragments or method chunks stored in a method repository - assembly-based. The methods constructed will be modular,

thanks to the interconnection between the fragments. That modularity leads to more flexible and adaptable methods [2].

## 1.1 Problem

In the context of Business Process Modelling, we can lead to a set of different BPD's and different views over that according to the business stakeholders' concerns. Those views are obtained from viewpoints, but we don't know how to construct viewpoints knowing their constituents only.

The SME is a methodology that focuses on the creation of the specific methods on the fly according to the ISD situation, instead of looking for universally applicable ones. Therefore using the SME to construct methods to obtain views over BPD's we may reach a manner to create views in a consistent way. In the addressed context we aim to prove if the SME a mechanism that allows us to generate views over the same Business Process Model in a consistent way.

Our work objectives is to prove that we can use the SME and all the concepts addressed to it, to build a chunk library with chunks of viewpoints and use it to represent the stakeholders' concerns, in a multi-stakeholder perspective. We are aiming also to prove that the library is extensible and adaptable.

## 1.2 Questions and hypothesis

With this work we reach to the following addressed questions and their hypothesis:

**Q1:** *Is it possible to specify a view based in a well-defined method?*

**H1:** We can create methods to obtain views, trough the decomposition of viewpoints using chunks.

**Q1/H1** Using chunks to specify views turns possible creating consistent views. A chunk is defined in SME as a constituent block of a method. If we can create views based in a method we obtain consistent views.

**Q2:** *Can we construct a library with viewpoint chunks?*

**H2:** The library can be constructed using the chunks created in H1 to populate the library.

**Q2/H2** The library makes reuse possible; serves as a repository for chunks; can be incrementally created; is a key artefact in SME.

**Q3:** *Is it possible to apply refactoring over a library of viewpoint chunks?*

**H3:** Use the chunks of H2 modifying them to reuse in the maximum number of viewpoints - situations.

**Q3/H3** Refining chunks we reduce granularity; more granular chunks leads to more reuse.

## 2 Related Work

As core supporting literature to the present work we based on SME and IEEE1471 key concepts, therefore through the next two sections will be briefly described those matters.

### 2.1 Situational Method Engineering

The discipline of SME focuses on the creation of the specific methods on the fly, instead of looking for a universally applicable ones [2]. Taking into account the project context and type, SME aims the process of designing, constructing and adapting methods for the development of Information Systems [3].

There are some approaches of SME but most of them promote de construction and adaptation of new methods by assembling reusable method fragments or method chunks stored in a method repository - assembly-based [2,4]. The methods constructed will be modular, thanks to the interconnection between the fragments. That modularity leads to more flexible and adaptable methods.

The work used key concepts of SME are:

- **Method Chunk Repository**, also presented as Chunk Library, will be a repository to store Method Chunks. In this work this is a conceptual artefact, it will not be constructed.
- **Method Chunk**, as mentioned on the Related Work chapter, a method chunk is a reusable building block to a method construction or adaptation as the situation requires. Also known as Situation Chunk.
- **Reuse Frame**, will be a table storing references to the chunks on the Chunk Library. Additionally it will be addressed an index with the situations referenced on the Reuse Frame.

### 2.2 IEEE 1471, Viewpoints and Views

In Enterprise Architecture practically no one except the architect is interested in a full architecture, in regards of full scope and detail. There will be clients and users of the system interested in specific parts of the architecture [6]. Therefore appears the concepts of Stakeholders, Concerns, View and Viewpoint, that the standard IEEE 1471 identifies [5, 6, 7, 8]:

- **Stakeholders** are the persons who have interests in an architectural part of a system.
- **Concerns** are the interests of the stakeholders. Therefore, concerns are simply things the stakeholders care about a system.
- **A View** is a representation of the system from the perspective of its related concerns;

- **A Viewpoint** specifies the conventions to construct and use a view. It is a pattern or a template to develop a view, establishing the view's audience and purposes.

A view will lead to a model or a set of models of the system's architecture and if a stakeholder requires information from aspects in different layers of the architecture, it will be integrated.

### 3 Method Proposal

To achieve our solution we have **two main methods**, the one who is responsible to the definition of our chunks and to populate the reuse frame addressed to the chunk library, with chunks of situations. The other method we will use to prove our solution putting in practice an approach to the Assembly-based Method applied to the SME. Supporting our work, the SME have some key concepts as mentioned on the Related Work chapter. The SME was previously applied to Organizational Engineering and Business Process (Re)Engineering adapting the major practices and concepts, and we based our proposed method on that approach [9].

#### 3.1 Method One – Chunks Creation and Managing Reuse Frame

The SME Reuse Frame will address all the situations that our library stores as well as all the chunks that the situations need. Thus we need to create our chunks and reference them in the Reuse Frame.

When we start to reference our first situation the Reuse Frame will be empty, but when we had situations populating it and we want to introduce a new one, it will be needed to find out if we can reuse other situations chunks or modify the chunk or chunks that we want to use of situations with similar purposes and constituents. In case of modification, the chunk will be refactored as we do with programming functions, to turn the Information Systems or programs, more extensible and modular. Thus the chunk(s) will not be addressed to one situation only.

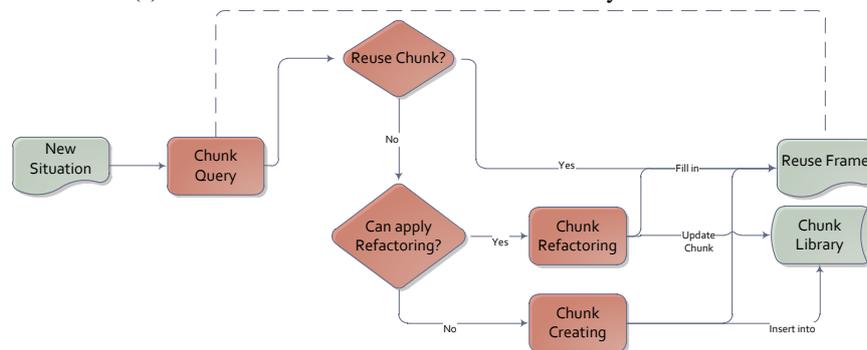


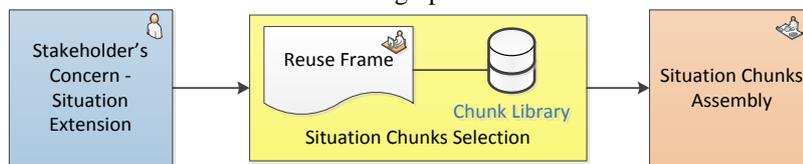
Figure 1: Proposed Method One

### 3.2 Method Two – Situation Assembly

Our method is settled in the assembly-based approach applied to the SME, where we can differentiate three main steps to reach the desired Situation.

To satisfy the stakeholders concerns we will identify the situation in the reuse frame, than we can retrieve our chunks from the chunk library to assemble them, obtaining the desired product - Figure 2.

The chunks assembling step is figurative because each situation solution in the Reuse Frame is represented by steps, and each step we will be represented as a graph node. Therefore when we select the chunks to prepare the solution pretended by the stakeholder we will start to construct a graph.



**Figure 2:** Situation Assembling Method

## 4 Applications

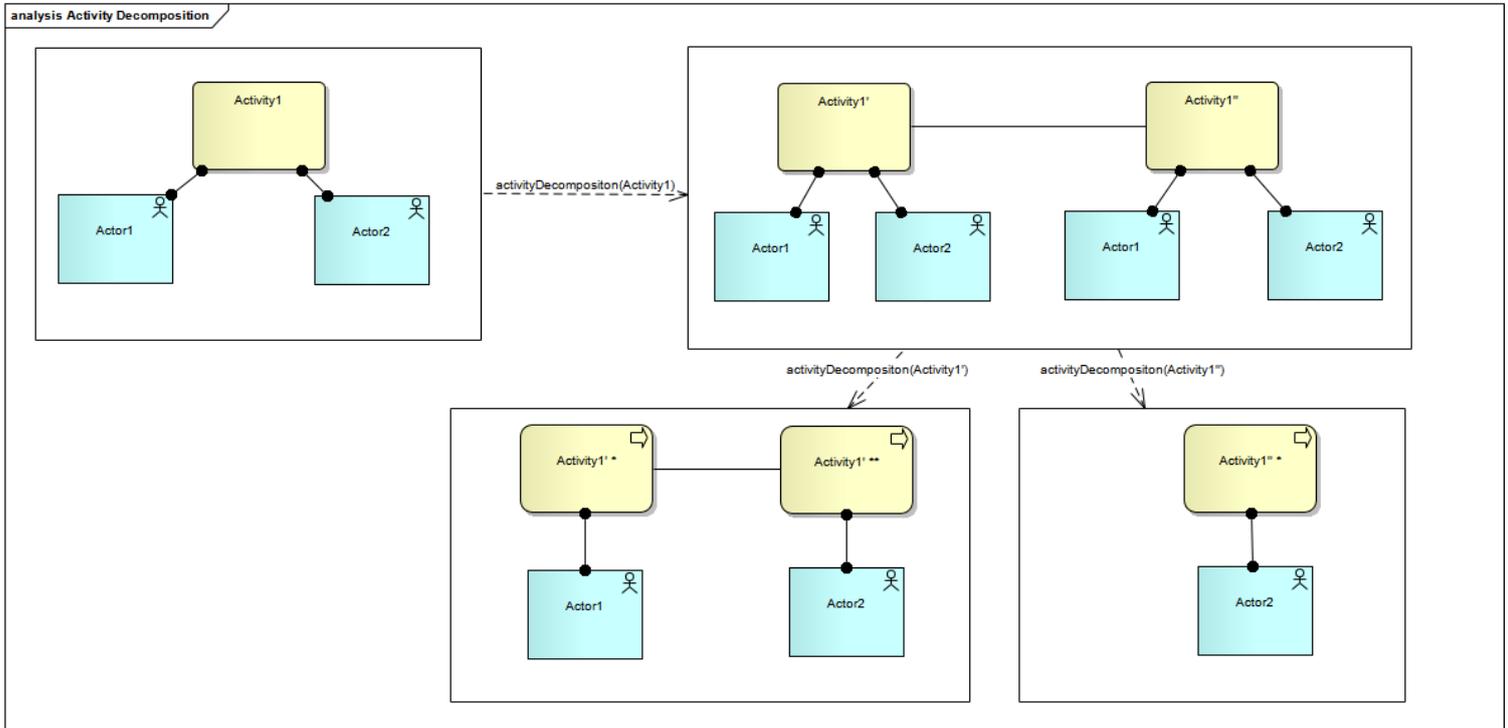
In the context of this work, we are about to prove the use of SME to obtain views over a model in a consistent way, to support that we propose two situations applying the proposed methods.

### 4.1 Situation One - Actor Cooperation Viewpoint

A business activity can be performed by actors from different Organisational Units (OU), in fact we have located a collaboration. But in this specific situation, a collaboration can be performed by actors from the same OU. Therefore we propose the following steps to identify collaborations between actors:

- 1) Internal Actors identification;
- 2) Identify which activities each actor executes;
- 3) Identify activities and relations between them;
- 4) Decompose activities while they are executed by more than one actor. When the activity decomposition stops, the cooperation will be represented in the step immediately before an activity or activities is being executed by one actor only.

Following the steps for this situation will be obtained the model represented by Figure 3. But our objective is to obtain a view for actor cooperation, thus as the last step indicates, the collaboration is one state before the last decomposition that present activities performed by one actor. That's the actor cooperation view.



**Figure 3:** Model for actor cooperation viewpoint

## 4.2 Situation Two - Context Diagram Viewpoint

As objective of this situation we aim to obtain a view with an enterprise context diagram.

We propose the following steps to the build a context diagram:

- 1) External Actors identification;
- 2) Identify which activity each external actor intervenes;
- 3) Identify activities and relations between them;
- 4) Decompose activities while they have relations with external actors. When the activity decomposition stops, the external relations will be in the step immediately before an activity has no relations with its environment.

The situation addressed above has similarities with S1 - Actor Cooperation Viewpoint. Beyond the possible common chunks, the one that has the instructions to decompose activities and identify actor cooperation can be refactored to identify relations with the organization environment (fourth chunk of S1).

## 4.2 Populating the Reuse Frame

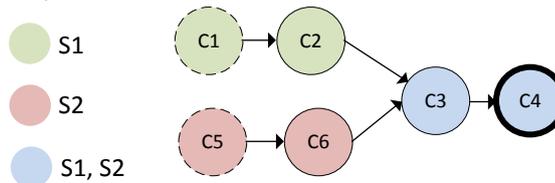
Finally we need to populate the already proposed SME Reuse Frame with the steps of the new situation – S2 Context Diagram Viewpoint, and the modifications to the fourth chunk of S1.

Situation	Chunk	Activities	Result
S1	C1	Internal Actors identification	All internal actors identified
S1	C2	Identify which activities each actor executes	Activities and actors relations
S1,S2	C3	Identify activities and relations between them	Activities and their relations
S1,S2	C4	Decompose activities while they are executed by more than one actor. When the activity decomposition stops, the cooperation will be represented in the step immediately before an activity or activities being executed by one actor only. Or decompose activities while they have relations with external actors. When the activity decomposition stops, the external relations will be in the step immediately before an activity has no relations with its environment	Actor cooperation view or a context diagram designed
S2	C5	External Actors identification	All external actors identified
S2	C6	Identify which activity each external actor intervenes	Activities and external actors relations

**Table 1:** Reuse Frame filled - Situation 1 and 2 of Applications Chapter

Choosing the chunks to obtain an actor cooperation view or a context diagram we lead to the following graph of activities, representing their execution order, helping us to assemble the situation chunks achieving the desired result.

Key:



**Figure 4:** Situation One and Situation Two - Activity Flow

## 5 Discussion and Final Remarks

In this work we have applied the Situational Method Engineering to obtain consistent methods to generate views over BPD's or other EA diagrams. So we proposed two methods - the one which allows us to make a heterogenic and extensible chunk library, where we can reuse and modify the chunks according the situations needs, and the other one which is based in the SME assembly-based methodology. Therefore the usage of SME allows us to generate views over the same Business Process Model in a consistent way. Using our proposed methods we can generate consistent viewpoints with well-defined guidelines.

To validate our solution basing on the solutions and situations presented along this work, we explain the answers that were proposed in first instance to the addressed problem.

**Q1:** *Is it possible to specify a view based in a method or mechanism?*

**H1:** Create methods to obtain views, decomposing viewpoints in chunks.

In chapter 4, presenting the situations One and Two, respectively on section 4.1 and 4.2, we proposed two methods leading to two different viewpoints. The structural parts of the viewpoints are the chunks we have made, representing steps to obtain the desired viewpoints. We have used the proposed method on section 3.1, to create the chunks and the method on 3.2 to assemble the chunks.

**Q2:** *Can we construct a library with viewpoint chunks?*

**H2:** Use the chunks created in H1 to populate the library.

We used the proposed method on section 3.1, starting with the reuse frame empty and adding situation chunks obtained on chapter 4.

**Q3:** *Is it possible to apply refactoring over a library of viewpoint chunks?*

**H3:** Use the chunks of H2 modifying them to reuse in the maximum number of viewpoints - situations.

With the obtained chunks on the situations of chapter 4, we modified and adapt them to fit both situations. As mentioned before we used our proposed method on chapter 3.1.

As main contributions of this work we present the following:

- The usage of SME adapted to viewpoint creation using the concepts of refactoring and reutilization;
- A method to obtain views and an incremental library;

This work guarantees that we may use a mechanism to obtain consistent views over BPD's or other diagrams over EA. The quality of the diagrams where we aim to obtain views is not relevant; the important is the final result, in other words, the obtained view.

## References

1. Lankhorst, M., et al.: Enterprise Architecture at Work, pp. 88—94. Springer, Heidelberg, ISBN 978-3-642-01309-6 (2009)
2. Mirbel, I., Ralyte, J.: Situational method engineering: combining assembly-based and roadmap-driven approaches. Springer, London (2005)
3. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering On the Differentiation of Context and Project Type. University of St. Gallen (2007)
4. Brinkkemper, S., Saeki, M., Harmsen, F.: Assembly Techniques for Method Engineering. Springer, Heidelberg (1998)
5. Hilliard, R.: All About IEEE Std 1471 (2007)
6. Lankhorst, M., et al.: Enterprise Architecture at Work, pp. 54—72. Springer, Heidelberg, ISBN 978-3-642-01309-6 (2009)
7. Hilliard, R.: Aspects, Concerns, Subjects, Views, ... \*. Integrated Systems and Internet Solutions, Inc., Massachusetts (1999)
8. Maier, M., Emery, D., Hilliard, R.: ANSI/IEEE 1471 and Systems Engineering. Systems Engineering, Vol. 7, No. 3 (2004)
9. Winter, R., Gericke, A.: On the Application of the ISD Method Engineering Approach to Organizational Engineering. Institute of Information Management University of St. Gallen (2009)