Aligning Business Processes and Software Connecting the UML 2 Profile for Event Driven Process Chains with Use Cases and Components*

Birgit Korherr and Beate List

Women's Postgraduate College for Internet Technologies
Institute of Software Technology and Interactive Systems
Vienna University of Technology
{korherr, list}@wit.tuwien.ac.at
http://wit.tuwien.ac.at

Abstract. The alignment between business processes and software is inadequately supported in conceptual modelling, although business processes are often the starting point for software development and define requirements for software systems. To address this gap, we connect the UML 2 profile for Event Driven Process Chains (EPCs) [2] with UML 2 elements representing software requirements and components. The approach is tested with an example business process.

1 Introduction

Business processes are often the starting point for software development and define requirements for the software systems to be designed. However, until now research and industry have only marginally addressed the alignment of business processes and IT. In order to address this gap, the goal of this paper is

- to conceptually describe the alignment between business processes and software by
- connecting the UML 2 profile for Event Driven Process Chains (EPCs) [5], introduced by the authors in [2] with UML 2 elements representing software requirements and components.

The UML 2 profile for EPCs represents a mapping from EPCs to UML 2 activity diagrams and aims at providing business process models to software developers in a well known notation. In this paper, the alignment is focused on the software requirements of a business process and the software components that are necessary to successfully implement and execute the process. The contribution of the alignment is:

- Business process models that are used as a starting point for software development support the achievement of a business goal-oriented software development.

* This research has been funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

- The alignment can provide a better description of a business process and its supporting software systems.
- Business process models in general and the UML 2 profile for EPCs in particular, can be utilised to elicit requirements for a new software system, but also for checking whether the functions of an existing software system match the requirements of a new business process.
- The UML 2 profile for EPCs connected with UML 2 elements representing software can be seen as a further step towards bridging the gap between business process engineering and software engineering.

The connection of the UML 2 profile for EPCs with UML 2 elements is described and illustrated with examples in Section 2. Related work is discussed in Section 3, followed by the conclusion in Section 4.

2 Connecting Business Processes with Software

In this section, we describe the details of how a business process, represented as a UML 2 profile for EPCs [2] can be aligned with software. Therefore, we connect business process activities, in our profile stereotyped actions, called *«elementary function»,* with the UML 2 elements *component* and *use case* by using *dependencies*. Use cases are suitable for defining software requirements, while components represent the modular structure of a software system. The Object Management Group (OMG) [4] describes use cases as a collection of actions, which stand for a specific behaviour. According to the OMG [4], a component covers physical and logical modelling aspects; this means that a component is a modular part of a system. Thus, use cases represent software in an abstract way, which means that no concrete implementation stands behind them, while components describe a software system or part of it.

We use dependencies to connect stereotyped actions with use cases and components, because with dependencies it is possible to connect UML 2 elements from behavioural diagrams with elements from structural diagrams. By contrast, associations cannot connect stereotyped actions with use cases and components, because the meta-class *property* represents the association end and belongs to the structural models, while the UML 2 profile for EPCs belongs to the behavioural models. Generally, in UML 2 it is impossible to link the two different modelling types with associations, but with a dependency it is possible to show that an element, called client, is dependent on another element, called supplier [4]. This means that a modification of the supplier may impact the client. For instance, the client might need the model element of the supplier for its specification or implementation. The graphical notation of a dependency is a dashed arrow. The model element at the tail of the arrow (the client) depends on the model element at the arrowhead (the supplier).

Figure 1 presents the *processing of automobile insurance claims* business process and its dependencies with stereotyped actions and use cases as well as software components. The process starts with the stereotyped action *record the claim,* which requires for its execution the component *claim management system* and the use cases *check policy* and *formulate claim description*. Furthermore, the claim management

system is also needed by the stereotyped actions *calculate the insurance sum* and *examination of results*. After the claim is recorded, the use case *proof of documents* is necessary for the calculation of the insurance sum. If the insurance sum represents a major amount, then the stereotyped action *checking history of the customer* is necessary. This action also requires the *customer relationship management (CRM) system*. Finally, if the action examination of results is positive, the *bank transfer* component is used by the stereotyped action to *pay for the damage*, otherwise the action *do not pay for the damage* is processed. The process ends with a closed case. With respect to the first part of the process, which is accomplished by the organisational role of the *financial claim specialist*, one can see that this model provides a very good impression of a process' software requirements and integrates a software perspective into the process model. The second part of the process illustrates which stereotyped action depends on a specific software component for its execution, and is fulfilled by the organisational role of the *claim administrator*.

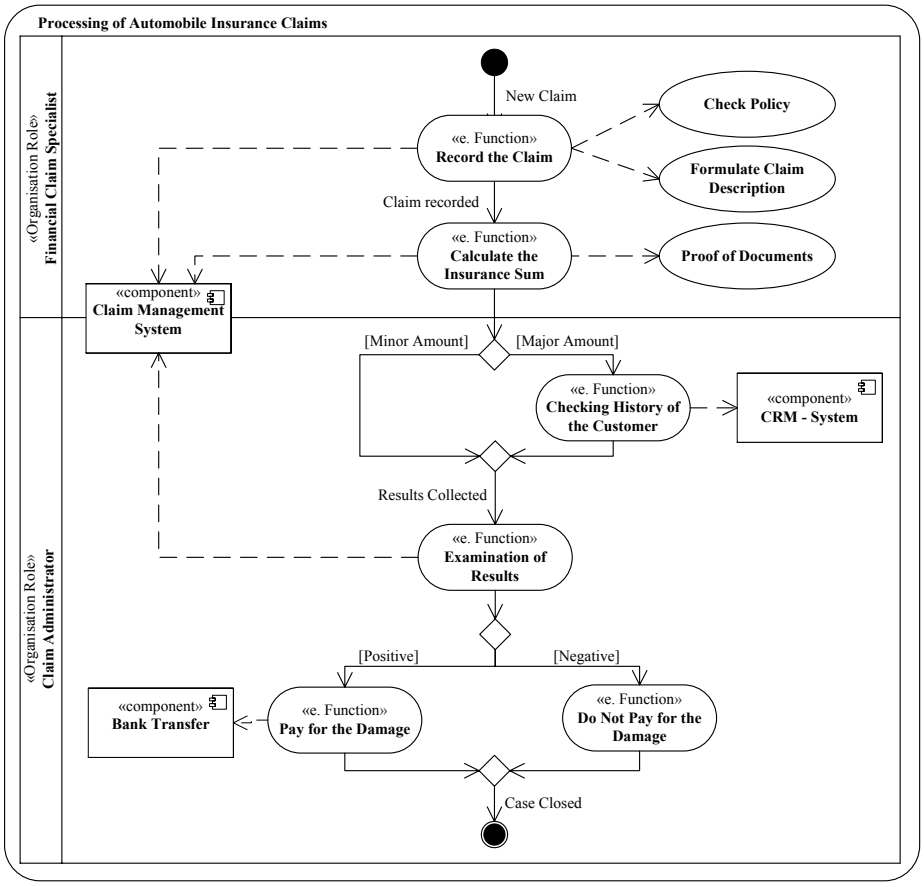

Figure 1: Processing of Automobile Insurance Claims Business Process

3 Related Work

There are quite a lot of conceptual Business Process Modelling Languages (BPMLs) and UML profiles for business process modelling available. They focus primarily on the sequential flow of the business process and do not integrate software elements [3]. An exception is the UML profile for Business Modelling in [6]. It views the integration of business processes and software development from an industry perspective. The profile maps business concepts to software artefacts.

The EPC [5] provides for its functions the possibility to access so-called information objects, which represent database tables or attributes, if more detailed is required.

In the Business Process Modeling Notation (BPMN) [1], data objects are considered as artefacts, which are documents, data, and other objects used and updated during the process. Data objects can be both electronic and physical. There is no explicit notation element for software available in the BPMN. Thus, the meaning of a data object is ambiguous like in many other BPMLs [6].

4 Conclusion

In this work, we connected the UML 2 profile for EPCs, introduced by the authors in [2] with UML 2 elements representing software, in order to support the alignment between business processes and software. We connected stereotyped actions with use cases as well as with components by the means of dependencies. Use cases address the elicitation of software requirements supporting an action. Components represent the software systems an action requires for its execution. The approach was tested with an example business process.

References

[1] Business Process Management Initiative: Business Process Modeling Notation, Specification Version 1.0, November 13, 2002, http://www.bpmn.org/

[2] Korherr, B. and List, B.: A UML 2 Profile for Event Driven Process Chains, Proceedings of the International Conference on Research and Practical Issues of Enterprise Information Systems (Confenis 2006), Vienna, Austria, IFIP Series, Springer Verlag, 2006.

[3] List, B. and Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages, Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06), April 2006, Dijon, France, ACM Press, 2006.

[4] Object Management Group, Inc.: UML 2.0 Superstructure http://www.omg.org/

[5] Scheer, A.-W.: ARIS – Business Process Modeling. Springer Verlag 1999.

[6] Tyndale-Biscoe, S., Sims, O., Wood, B., Sluman C.: Business Modelling for Component Systems with UML. Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC '02), IEEE Press 2002.