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

Birgitt Korherr, Beate List
Women’s Postgraduate College for Internet Technologies
Institute of Software Technology and Interactive Systems
Vienna University of Technology
{korherr, list}@witt.tuwien.ac.at, http://witt.tuwien.ac.at

Overview

Motivation
- Business processes are often the starting point for software development and define requirements for software systems.
- The alignment between business processes and software is inadequately supported in conceptual modelling [1].

Goals
- To conceptually describe the alignment between business processes and software by connecting the UML 2 profile for Event Driven Process Chains (EPCs) [2] with UML 2 elements representing software requirements and components.

Contribution
- To support a business-oriented and goal-oriented software development
- To provide a better description of a business process and its supporting software systems
- To bridge the gap between business process engineering and software engineering

Why a UML Profile for Event Driven Process Chains (EPCs)?

- EPCs have become widely used for business process modelling in organisations
- EPC can be used as a starting point for software development
- But most software developers are not aware of EPCs
- To provide EPC models to software developers in a well-known notation
- To present EPC models to software developers through UML tools

Fig. 1. Example of a mapping from EPCs to the UML profile for EPCs

Connecting the UML Profile for EPCs with Software Elements

Aligning the UML 2 profile for EPCs with Software
- Connecting 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
- Components represent the modular structure of a software system

Dependencies ...
- ... connect stereotyped actions with use cases and components
- ... show that an element, called client, is dependent on another element, called supplier

Example of a dependency relationship between an "elementary function" and a use case:

Client  
<table>
<thead>
<tr>
<th>Dependency</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g., Function</td>
<td>Internet Buying</td>
</tr>
<tr>
<td>Credit Card Payment</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2. The "elementary function" Internet Buying requires the use case Credit Card Payment for its execution.

Example of a dependency relationship between an "elementary function" and a component:

Client  
<table>
<thead>
<tr>
<th>Dependency</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g., Function</td>
<td>Book Concert Tickets</td>
</tr>
<tr>
<td>Ticket Management System</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 3. The "elementary function" Book Concert Tickets needs for its full implementation the component Ticket Management System.

Example Business Process

Example business process consists of two parts:
1. First part of the process is accomplished by the organisation role of a financial expert
   - The stereotyped action record the claim requires for its execution the component claim management system and the use case check policy and formulate claim description
   - After the claim is recorded, the use case proof of documents and the component claim management system is needed by the stereotyped action calculate the insurance sum
2. Second part of the process is accomplished by the organisation role of the employee of the financial department
   - If the calculated insurance sum is high, then checking history of the customer is necessary
   - Checking history of the customer requires the customer relationship management (CRM) system
   - Examination of results needs the component claim management system
   - Examination of results is positive: bank transfer component is used by the stereotyped action pay for the damage
   - Examination of results is negative: e.g. function do not pay for the damage will be used
   - The process ends with a closed case

Fig. 4. Example of a business process for Dealing with Automobile Insurance Claims

Results

- Model provides a very good impression of a process' software requirements
- Model integrates a software perspective into the process model
- Can be utilized to elicit requirements for a new software system
- Use cases address the elicitation of software requirements supporting an action
- Model illustrates which stereotyped action depends on a specific software component for its execution

References


Acknowledgement

This research has been funded by the Austrian Federal Ministry for Education, Science, and Culture, and the European Social Fund (ESF) under grant 31.963/84-1/9/1002.