Using Domain Specific Languages for Software Process Modeling

Darío Correal   Rubby Casallas
{dcorreal,rcasalla@uniandes.edu.co}

The 7th OOPSLA Workshop on Domain-Specific Modeling

Montreal - Canada, October 21-22, 2007
Outline

1. Problem Domain
2. Proposal
3. Discussion
Problem Domain

Software Process Modeling

- Activities
- Control Flow

- CreateDLD
- Codification
- UnitTesting
- CreateBL
- RegisterDefects
Business Process Modeling

Defining and Maintaining a Process Definition is a Complex task

- Multiple activities
- Multiple stakeholders
- Multiple concerns
Business Process Modeling

Defining and Maintaining a Process Definition is a Complex task

- Multiple activities
- Multiple stakeholders
- Multiple concerns
Defining and Maintaining a Process Definition is a Complex task

- Multiple activities
- Multiple stakeholders
- Multiple concerns
Example

Design and Codification of a Software Component

- There are multiple concerns defined around this process
- Stakeholders expect this resulting process !!!
**Design and Codification of a Software Component**

- There are multiple concerns defined around this process
- Stakeholders expect this resulting process!!!
Design and Codification of a Software Component

- There are multiple concerns defined around this process
- Stakeholders expect this resulting process !!!
Main Objectives:

- To make process definitions explicit with the use of models
- To reduce the complexity of changing process models
Main Objectives:

- To make process definitions explicit with the use of models
- To reduce the complexity of changing process models
A *target process* (TAP) is a business process selected for improvement inside an organization.
A viewpoint [10] is the perspective from which a particular stakeholder interacts with a TAP in terms of the concerns relevant to him.
Our Strategy is organized in the following way

1. Target processes definition using process models
2. Viewpoint definition using a Domain-Specific language
3. Target process and Viewpoints Integration
1. Target Process Definition

We defined a Business Process Metamodel

- Based on the Business Process Modeling Notation (BPMN) specification proposed by the OMG [1]
2. Viewpoint Definition

We designed and defined the AspectViewpoint Language

- A Domain-Specific Aspect Language (DSAL)
- To create Viewpoint models (Concerns)
- Uses the Workflow Control Patterns [11] as part of its vocabulary
2. Viewpoint Definition (Cont.)

With AspectViewpoint we can express:

- Where the viewpoint affects the target process
- How the viewpoint modifies the target process
With AspectViewpoint we can express

- Where the viewpoint affects the target process
- How the viewpoint modifies the target process
2. Viewpoint Definition (Example)

```java
Strategy addReview (Task task) : after findCodification (task) {
  Task inspection
  inspection = task.parent().newFlowObject("srcReview,"Task", "Normal");
  task.Sequence ( [inspection,"" ] );
}
```

![Target Process Diagram]

CreateDLD → Codification → UnitTesting → CreateBL

CreateDLD → RegisterDefects → Codification
2. Viewpoint Definition (Example)

```java
Strategy addReview (Task task) : after findCodification (task) {
    Task inspection
    inspection = task.parent().newFlowObject("srcReview","Task", "Normal");
    task.Sequence ( [inspection,""] );
}
```
2. Viewpoint Definition (Example)

```
Strategy addReview (Task task) : after findCodification (task) {
    Task inspection
    inspection = task.parent().newFlowObject("srcReview,"Task", "Normal");
    task.Sequence ( [inspection,""] );
}
```
3. Target Process and Viewpoints Integration

Woven Process Model


Viewpoint Model

ECL [5] Strategies
4. Technologies Used

- **Generic Modeling Environment (GME) [9] - BPMN metamodel**
  - OpenArchitectureware framework [8]
    - AspectViewpoint grammar definition
    - AspectViewpoint Metamodel generation
    - Transformation of Viewpoint models into ECL code

- **Embedded Constraint Language (ECL) [5] - Express**
  - Process Model transformations

- **C-SAW weaving engine [7] to transform process models**
4. Technologies Used

- Generic Modeling Environment (GME) [9] - BPMN metamodel
- OpenArchitectureware framework [8]
  - AspectViewpoint grammar definition
  - AspectViewpoint Metamodel generation
  - Transformation of Viewpoint models into ECL code
- Embedded Constraint Language (ECL) [5] - Express Process Model transformations
- C-SAW weaving engine [7] to transform process models
4. Technologies Used

- **Generic Modeling Environment (GME) [9]** - BPMN metamodel
- **OpenArchitectureware framework [8]**
  - AspectViewpoint grammar definition
  - AspectViewpoint Metamodel generation
  - Transformation of Viewpoint models into ECL code
- **Embedded Constraint Language (ECL) [5]** - Express
  - Process Model transformations
- **C-SAW weaving engine [7]** to transform process models
4. Technologies Used

- Generic Modeling Environment (GME) [9] - BPMN metamodel
- OpenArchitectureware framework [8]
  - AspectViewpoint grammar definition
  - AspectViewpoint Metamodel generation
  - Transformation of Viewpoint models into ECL code
- Embedded Constraint Language (ECL) [5] - Express
  - Process Model model transformations
- C-SAW weaving engine [7] to transform process models
4. Technologies Used

- Generic Modeling Environment (GME) [9] - BPMN metamodel
- OpenArchitectureware framework [8]
  - AspectViewpoint grammar definition
  - AspectViewpoint Metamodel generation
  - Transformation of Viewpoint models into ECL code
- Embedded Constraint Language (ECL) [5] - Express Process Model transformations
- C-SAW weaving engine [7] to transform process models
5. Experimentation

- We work with a Software Development Company
- Technical solution process area - CMMi level 1 (Continuous Representation)
- We created a Target Process model to represent this process
- The objective was to modify the process to achieve level 2
- The test cases in this scenario showed how the generic practices (concerns) were easily modeled using our viewpoint language
- After the integration of the TAP and the Viewpoints we obtained the expected processes.
Conclusions

1. We reduced the complexity of changing process models
   - We provided a Domain-Specific model for representing stakeholder’s concerns
   - In a Modular and non-intrusive way

2. Stakeholders and process designers improved their communications using the AspectViewpoint language
Discussion

- Questions
- Comments
Bibliography I


Bibliography II

Yuehua Lin, Jeff Gray, and Jing Zhang.  
The embedded constraint language: A transformation language for visual models.  
Submitted for publication, 2006.

Kapsammer Elisabeth Retschitzegger Werner Wimmer Manuel Schauerhuber Andrea, Schwinger Wieland.  
Towards a common reference architecture for aspect-oriented modeling.  
In 8th International Workshop on Aspect-Oriented Modeling, in conjunction with AOSD’06, Bonn, Germany, March 21., 2006.

SoftCom.  
C-saw.  
http://www.gray-area.org/Research/C-SAW/.

SoftCom.  
OpenArchitectureware  
http://www.openarchitectureware.org/.
Akos Ledeczi.  
The generic modeling environment.  

Ian Sommerville, Gerald Kotonya, Steve Viller, and Peter Sawyer.  
Process viewpoints.  

W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros.  
Workflow patterns.  