

Flow of Work

Ideas for Process Modeling and Documentation

Author: Thomas Schuetz, Protos Software GmbH

1 Introduction

Developing and documenting processes is an important but time consuming task. Using a model driven approach can help to make this work more productive and less error prone.

1.1 Existing Process Modeling Tools

The existing commercial and open source modeling tools available are either very complex or are lacking functionality or both. Most tools are based on standards like SPEM and BPMN which leads to big, complex tools.

1.2 Goal

With *Flow of Work*, we want to create a lightweight process modeling tool to define processes and generate different documentation views.

The Initiators of this project work in the domain of software engineering, but the tooling can also be used to model processes in other domains.

1.3 Basic Principles

- **No Redundancy:** All information is single source. Each modeling element (Activity, Artifact, Document, Role and so on) will be modeled once and only be referenced from other elements thus enforcing the consistency of model changes.
- **Textual Syntax:** A purely textual syntax is used to build the models (a Domain Specific Language or DSL). This enables the fast development of the tooling and has lots of benefits regarding the configuration management of the models (versions, diff, merge).
- **Document Generation:** All Documents are generated automatically from the models.

2 Modeling Elements

2.1 Activity

- An *Activity* is the basic building block for the description of processes.
- *Activities* can be composed hierarchically.
- *Activities* can have input and output interfaces typed by *WorkProducts* and their state

2.2 WorkProduct

- A *WorkProduct* represents information that flows through the process and its single *Activities*.
- A *WorkProduct* can be created, consumed or transformed (change of state) by an *Activity*.
- A *WorkProduct* can define the type of an input or output of an *Activity*.

2.3 Role

- A *Role* defines who is responsible for an *Activity*.
- A Role is usually responsible for many *Activities*.
- Examples: Project Manager, Software Developer, Requirement Engineer

2.4 Guidance

- A *Guidance* defines additional Information or Items needed for an *Activity*.
- Examples: Templates, Checklists, Tools

3 Languages

3.1 Konkrete textual Syntax

- The konkrete textual syntax is used to create and edit the process model.
- The konkrete textual syntax of the language of Flow of Work is implemented in Xtext, which enabled the creation of state of the art text editors with syntax highlighting, code completion, and so on.

3.2 Konkrete graphical Syntax

- For some views the OMG process modeling standard SPEM will be used to create graphical diagrams for the process models.
- Not all concepts of the language of *Flow of Work* map to SPEM diagrams, but we will try to keep the diagrams similar.

4 Views

4.1 Navigation

- All views will contain useful links to other views of referenced modeling elements als links.
- The toplevel line of each view will contain a navigation bar with the navigation path.

4.2 HTML Views

For the first version we will provide only generated HTML views.

4.2.1 List of Views

4.2.1.1 Overview Views

- **All Activities:** List of all *Activities*
- **All Roles:** List of all *Roles*
- **All Work Products:** List of all *WorkProducts*

4.2.1.2 Detail Views

- **Activity description:** description of an Activity (TBD: HTML View for Activity?)
- **Role description:** description of a Role (TBD)
- **Work Product description:** description of a Work Product (TBD)

- **Hierarchical Activities:** complete *Activity* hierarchy (TBD)
- **Process Step:** only information important for the developer (without standard, ...) (TBD)
- **Prozess description view:** (Test, Projectmanagement, Supplier Management) (TBD)

4.2.2 Specification of HTML Views

4.2.2.1 HTML View for Activity

The HTML view for an *Activity* will be the first to be implemented.

Basic Information			
Purpose:	<i>Why is the step needed and for what -> motivation, explanation of the process step in itself</i>		
Scope:	<i>Which disciplin, e.g. Hardware development, Production, Sales,... or organizational scope, e.g. only LXS software development</i>		
Inputs:	<i>Documents / Information within a tool, e.g. change request</i>	Outputs:	<i>Documents and their status (draft, released, ready for review) or Information and status</i>
Entry Criteria:	<i>Criteria which have to be fulfilled before this activities may be started</i>	Exit Criteria:	<i>Conditions which have to be fulfilled to exit this process step</i>
Responsible:	<i>The role who is responsible that the results of this process brick are accomplished, e.g. software project manager</i>		
Stakeholder:	<i>Responsible for the description of this process brick</i>		
Description			

Activities and Responsibilities

Activities	Responsibilities/Roles
<i>Description of the activity</i>	<i>Role which has to do the activity RASI. For responsible roles put (R), approve (A), for supporting roles put (S), informed (I)</i>
...	...

If necessary a flow chart picture can be introduced here.

Notes

Additional not process relevant information which might be helpful, e.g. "a good example for a reference document you can find under the following link"

Implementation support

Templates

Empty templates which should be used within this process step.

Guidelines

Additional explanation on how to produce work results of this step, e.g. a guideline on how to do FMEAs. References to the PEP should be also mentioned here.

Tools

e.g. DOORS, MKS, Enterprise Architect

Trainings

Process training material

References

Standards

Standards which are considered at this step including the information about the corresponding part, e.g. "ISO26262-3 5.1 The first objective of the item definition is to define and describe the item. The second objective is to support an adequate understanding of the item so that each activity defined in the safety lifecycle can be performed."

To be filled by process experts CMMI, SPICE, ISO 26262

Internal documents

e.g. P177001 Embedded Software Process

Additional information (Literature)

e.g. eXtreme Programming

4.3 Dokument Generators

It is possible to generate also other formats of documents (e.g. PDF). This will be part of later releases.