REPROTOOL Workflow

(Textual documents in SW development)

D3S Seminar 2011-04-27

http://d3s.mff.cuni.cz

Viliam Šimko
simko@d3s.mff.cuni.cz
Textual documents in SW development

Textual Specifications  Manual error-prone translation  Formal Models

See Example SRS document
Our Vision: Textual documents in SW devel.

Benefits:

- Updating model from text
  - The customer may introduce changes in textual form
- Updating text from model
  - Textual presentation of formal models in customer's own words
- Using the extracted semantics
  - Consistency checking among several use-cases
  - Traceability of requirements to use-cases
  - Coverage of requirements

Specification as a structured text linked with models (improved specification)

Original Text

Initial import (customer's specification)

Tagged Text

Models

Semi-automatic inference of elements (Actors, Requirements, EFPs, Use-cases ...)

REPROTOOL
Reprotool Workflow Overview

1. Importing initial version of a document (structure of the text)
2. Model inference (using NLP tools)
3. Model editing (e.g. UML editor)
4. Model-to-text (tagged document)
5. Customer proposes changes to the specification (preserving the tags)
6. Merging text changes back to the model (using tags)
Importing Initial Version of a Document

1. Importing initial version of a document (structure of the text)
Importing Initial Version of a Document

- The original document is written by a business analyst as a **plain text file** (for simplicity)
- Splitting the text into basic document elements (sections, paragraphs, sentences...)
- Initial model is created (structure without semantics)

Global Personal Marketplace
System Requirements Specification (SRS)
Version 1.0

Functional Requirements

The section of the SRS specifies the functional requirements of the GPM in terms of use cases and their associated use case paths.

Accountant

The subsection specifies the functional requirements primarily associated with accountants.

...
Inference of model elements from text

1. Original Text

2. Model inference (using NLP tools)
Inference of model elements from text

- The tool helps to keep traces between original text fragments and semantical model elements
- Deriving semantics using NLP tools:
  - Automatic detection of potential SE-related artefacts
  - Manual refinement of the tool's proposals
Editing the model

Model editing (e.g. UML editor)
Example: Editing the model

- Software engineer works on the model level e.g.:
  - Concretization of use-cases
  - Concretization of EFPs

```plaintext
:UseCase
Name = Accountant Updates Fee Schedule

:UseCase
Name = Introducing new use-case

:UCStep
The accountant sends a request to update the fee schedule to ...

:UCStep
The GPM sends the current fee schedule containing the mand...

:UCStep
Introducing new use case step.
```

(reordering)
Example: Fragment of our meta-model
Model-to-text transformation

1. **Tagged Text**
2. **Models**
3. **Original Text**
4. Model-to-text (tagged document)
Introduction
The section introduces the system requirements specification (SRS) for...

Use-case: Accountant Updates Fee Schedule

Use-case steps:
1. The accountant sends a request to update the fee schedule to...
2. The GPM sends the current fee schedule containing the mand...
Customer proposes changes to the specification (preserving the tags)
Communication with the customer

Tagged Text

Use-case: Accountant
Updates Fee Schedule

Use-case steps:

1. The ac...
2. The GP...

Tagged text is given to the customer

Special text editor for editing the tagged text preserving tags

Regular text editor can also be used if the enclosing tags are not affected

Changing the text within a container.

Reordering sub-containers

Deleting a container
Merging text changes back to the model

Tagged Text

Models

Original Text

Merging text changes back to the model (using tags)
Merging text changes back to the model

Modified Tagged Text

Introduction
The section introduces the system specification (SRS) for ...

Use-case: Accountant
Updates Fee Schedule

Use-case steps:
1. The GP...
2. The ac...

Generated Tagged Text

Introduction
The section introduces the system requirements specification (SRS) for ...

Use-case: Accountant
Updates Fee Schedule

Use case steps:
1. The ac...
2. The GP...

Even more text
Merging text changes back to the model

- Changes made by the customer can be easily merged back to the model.
  - Affected model elements are identified
  - Software engineer modifies the affected model elements
Reprotool Workflow Summary

1. Importing initial version of a document (structure of the text)
2. Model inference (using NLP tools)
3. Model editing (e.g. UML editor)
4. Model-to-text (tagged document)
5. Customer proposes changes to the specification (preserving the tags)
6. Merging text changes back to the model (using tags)
History of the tool

- **Procasor:**
  - Supervised creation of use-cases
  - Use-case steps sentences analyzed by NLP tools
  - Generated behaviour protocols (pro-cases)
    - Formal notation describing all the possible execution paths defined by use-cases
Future of the tool

- Requirements Processing tool
  - The goal is to bind specification documents with models (and code)
- Based on Eclipse
- EMF for handling models