SOFTWARE REQUIREMENTS SPECIFICATION

for

XModeler

Release 1.0

Version 1.0 approved

Contents

Revision History

1 Requirements

Req-1.0.0.0.1	Some requirements may be optional. Spec-2.0.0.0.1

1.1 MeMo-Language

Req-1.1.0.0.1	The MultiLevel language MeMo requires...
Req-1.1.0.0.2	The MultiLevel language MeMo requires...
Req-1.1.0.0.3	The MultiLevel language MeMo requires...

1.2 Kernel

1.3 X-Modeler

1.3.1 Diagrams

1.3.1.1 Multi-Level Diagram

1.3.2 Forms

1.3.2.1 FormsClient

Req-1.3.2.1.1	The FormsClient receives messages to add components.
Req-1.3.2.1.2	The FormsClient must be able to display these components: • Label • Textfield • Textarea • Checkbox The component is determined by the type of the value.
Req-1.3.2.1.2a	A boolean is shown as a Checkbox
Req-1.3.2.1.2b	An enum is shown as a drop-down-list
Req-1.3.2.1.2c	A short(?define) Strings or a number is shown as Textfield
Req-1.3.2.1.2d	A long(?define) String is shown as Textarea
Req-1.3.2.1.3	Labels can be grouped with other components to form a key-value-pair.
Req-1.3.2.1.4	A form client has a listener for changes which will be transferred instantly to the underlying model. There are no Save, Cancel or Undo buttons.
Req-1.3.2.1.5	The FormsClient is linked to the underlying model.
Req-1.3.2.1.6	The FormsClient's components are linked to the underlying model's parts by an id.
Comment	Labels should not be empty. Git Issue #34
Comment	Double click should not freeze the form. Git Issue #33

2 Specifications

Spec-2.0.0.0.1	Some specifications may be optional. Req-1.0.0.0.1

2.1 XMF-Language

2.2 Kernel

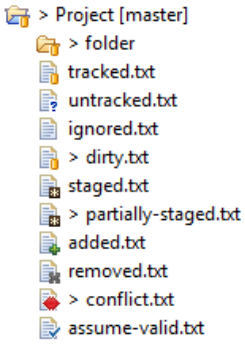
2.3 X-Modeler

2.3.1 Diagrams

2.3.1.1 Multi-Level Diagram

3 Other

3.1 git



- **dirty (folder)** - At least one file below the folder is dirty; that means that it has changes in the working tree that are neither in the index nor in the repository.
- **tracked** - The resource is known to the Git repository and hence under version control.
- **untracked** - The resource is not known to the Git repository and will not be version controlled until it is explicitly added.
- **ignored** - The resource is ignored by the Git team provider. The preference settings under Team > Ignored Resources, "derived" flag and settings from .gitignore files are taken into account.
- **dirty** - The resource has changes in the working tree that are neither in the index nor in the repository.
- **staged** - The resource has changes which have been added to the index. Note that adding changes to the index is currently possible only in the commit dialog via the context menu of a resource.
- **partially-staged** - The resource has changes which are added to the index and additional changes in the working tree that neither reached the index nor have been committed to the repository. See partial staging from the Git Staging view for how to do that.
- **added** - The resource has not yet reached any commit in the repository but has been freshly added to the Git repository in order to be tracked in future.
- **removed** - The resource is staged for removal from the Git repository.
- **conflict** - A merge conflict exists for the file.

- **assume-valid** - The resource has the "assume unchanged" flag. This means that Git stops checking the working tree files for possible modifications, so you need to manually unset the bit to tell Git when you change the working tree file. Also see Assume unchanged action.

3.1.1 fetch

Fetch from remote repository

3.1.2 pull

Incorporates changes from a remote repository into the current branch. In its default mode, git pull is shorthand for git fetch followed by git merge FETCH_HEAD.

3.1.3 commit

Commit changes to local repository

3.1.4 push

Push Commits to remote repository

3.1.5 conflicts

http://wiki.eclipse.org/EGit/User_Guide#Resolving_a_merge_conflict

3.1.6 branches

branch merge rebase cherry-pick

3.1.7 revert

If a commit which has not yet been pushed has to be undone:

Select folder Team → Show in History
Note number of commit
then use command line
`git revert <noted number>`

If a commit which has already been pushed has to be undone:

Is done the same way.

Note: Both reverts don't undo history. The wrong commit and the reversion will be logged.

4 Literature

- A Unifying Approach to Connections for Multi-Level Modeling [1]
  - Melanie: Multi-level Modeling and Ontology Engineering Environment [2]
  - Referenced 2
  - Referenced 3

5 Weekly Reports

5.15 2015

5.15.37 7 Sep 2015 - 11 Sep 2015

- Made browser tabs closeable
- Init Requirements and Specifications in LaTeX
- Added Panic button to GUI
- Image loading: XML file now derived from IMG file name and path instead of storing that information in the IMG file.
- Division of BigIntegers fixed, Multiplication of negative Integers fixed

5.15.38 14 Sep 2015 - 18 Sep 2015

- Tested and documented some git features.
- Tried to install safari. Waiting for Help desk...
- Made the browser work again.

5.15.39 ToDo

- improve webpage
- Specify FormClient
- git commands in Eclipse
- (nextWeek) MeMo language requirements
- pw protected folder: https://wincent.com/wiki/git_repository_access_control
- (Friday) produce conflicts in git
- Snippets

  HTNLViewer.xmf: Are ".o", ".xip", ".xto", ".xtd", ".xtml" still in use?

  Welcome.xmf: requestURL...

  requirements

Bibliography

[1] C. Atkinson, R. Gerbig and T. Kühne. A unifying approach to connections for multi-level modeling. *Proceedings of 18th ACM/IEEE International Conference MODELS*, 2015.

[2] C. Atkinson and R. Gerbig. Melanie multi-level modeling and ontology engineering environment. 2012.

[3] H. J. Farnsworth and L. Simpson. Mathematics in the year 3000. *Proceedings of 318th IEEE International Conference*, 55(4), 3015, pp. 309–319.