Software Requirements Specification

for

Metamodel tool

Version 1.0 approved

vrchlpet

MMN group

28.9.2010

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This project covers the needs of Independent modeler (IM) to provide a tool (plugin for IM), which could be able to design a new plugin for IM allowing to design an appropriate diagram (such as use case diagram).

1.2 Intended Audience and Reading Suggestions

This document is intended for any person who want to help us with development and MMN´s users. I suggest you should read all of it.

1.3 Project Scope

Project include development of plugin for IM, which will be able to create a source code of plugins for IM. Such a plugin will offer a toll for modeling certain diagram.

1.4 References

Other useful stuff can be found on https://rabbit.felk.cvut.cz/trac/MMN

2. Overall Description

2.1 Product Perspective

This project is about developing new plugin for Independent modeler. It is not a replacement of an existing plugin, it is an enhancement of IM. The plugin is intended to alleviate a development of new plugins for IM. The plugin will create a basic source code, which can be further edited.

2.2 Product Features

The plugin allow to design a metamodel of wide set of diagrams. This metamodel will be used to generate source code in java, which is basically a source code of plugin for IM. After adding this new plugin a user will be able to draw up an appropriate diagram in IM environment. Metamodels can be saved or loaded for further development.

2.3 User Classes and Characteristics

This plugin can be used by large variety of developers. For example, any developer making some effort to create a new plugin can find the MMN useful. It is much more easier if the developer can

design the shapes and its connections to each other in GUI and then let the MMN generate appropriate source code (basic sketch, which can be very easy to modified it later).

2.4 Operating Environment

With regard to a fact, that it is a plugin for IM, MMN can be used only in IM environment. IM is programmed in java so it can run anywhere, where is JRE installed (considering PC only).

2.5 Design and Implementation Constraints

Only thing, that could be count as a constraints is a fact, that MMN have to use an existing interface provided by IM.

3. System Requirements

Below you can find a table with MMN requirements specification

Non-Functional requirements		
requirement	**description**	**priority**
1	mmn will be programmed in java	**must have**
2	the source codes of new plugins will be generated in java	**must have**
3	mmn is a modul for Netbeans platform	**must have**
Functional requirements		
1	mmn will provide painting tools for creating new metamodels	**must have**
1.1	mmn will provide set of basic shapes such as rectangle or cicle	**must have**
1.2	shapes can be draw on workspace by DrugAndDrop technology	**want to have**
1.3	mmn will allow editing of shapes already laid on workspace	**must have**
1.3.1	in that case, only the connection bethween the shapes will remain without changes	**should have**
1.3.2	all settings of particular shaps will be accessible from dockable window	**must have**
2	mmn will allow to define connections between shapes	**must have**
2.1	mmn will provide set of basic connection types such as association or generalization	**must have**
2.1.1	mmn will allow to determine the direction of association	**must have**
2.1.2	mmn will allow to determine the multiplicity on both sides of association	**must have**
2.1.3	mmn will allow to chose between one-side or two-side association	**must have**
2.1.3	mmn will allow to determine the direction of	**must have**

	generalization	
2.2	precisely two shapes will be assigned to each connection	**must have**
2.2.1	one-side association will be marked with arrow pointing to the target	**should have**
2.2.2	two-side association wont be marked with any special sign, it will be displayed by straight line	**should have**
3	mmn will allow to define shapes attributes	**must have**
3.1	it will be possible to add or edit attributes by dockable window	**should have**
3.1.1	it will be possible to add an inicialization value to an attribute	**must have**
3.2	it will be possible to remove an existing attribute by dockable window	**should have**
3.3	a random number of attributes can be assign to a random shape	**must have**
4	mmn will generate XML file containing description of shapes and its connections	**must have**
5	mmn will allow to enhance already created metamodels	**must have**
5.1	mmn will allow to save metamodels	**must have**
5.1.1	during the saving process, mmn will demand basic values	**must have**
5.1.1.1	it is a file name	**must have**
5.1.1.2	and location	**must have**
5.2	mmn will allow to load an existing, previously saved metamodel	**must have**
6	mmn metamodels will be saved in XML file	**must have**
7	mmn will create an interface, that allow to use objects according to metamodel	**must have**
7.1	mmn will provide access point used to provide a pallete of defined shapse, this shapes will be used by other plugins through this access point, it is basicaly centralized repository for shapes definition	**must have**

Appendix A: Glossary

MMN – metamodeovací nástroj
IM – Independent modeler

Appendix B: Analysis Models

Can be found on https://rabbit.felk.cvut.cz/trac/MMN/attachment/wiki/Files/analysis%20model.jpg