Software Patterns
Requirements Definition

Version 1.0

Revision History

Date	Version	Description	Author
2010-10-01	1.0	Initial draft	Pađen, Ramljak, Grbić, Vitas

Table of Contents

1.Introduction

1.1Purpose of this document

This document has the purpose of elaborating requirements for the Software Patterns project. By defining these requirements there is little space left for misunderstandings of the project goals. Also, this document will represent reference during rest of the project's phases.

1.2Intended Audience

This document is of concern to:

- •Project team members - for implementation of the defined requirements
- •Customers - for disposing and overview of the requirements
- •Supervisors - for supervision of the project's progress

1.3Scope

Scope of this document contains:

- •Requirements definition and description
- •Use case models
- •Prediction of the future model

1.4Definitions and acronyms

1.4.1Definitions

Keyword	Definitions
Eclipse	Integrated development environment primarily used for development of Java applications.
Software pattern	Reusable solution to a commonly occurring problem in software design.
Plug-in	Set of software components that adds specific capabilities to a larger software application
Catalog	Term catalog is used in this context as a collection of categorized software patterns
JUnit	Unit testing framework for Java programming language

1.4.2Acronyms and abbreviations

Acronym or abbreviation	Definitions
IDE	Integrated Development Environment
UML	Unified Modeling Language
EMF	Eclipse Modeling Framework
EclEMMA	Eclipse plug-in that generates code coverage reports and provide simple trace information about test cases.
LaTeX	Document preparation system for high-quality typesetting most often used for medium-to-large technical or scientific documents.
UC	Use case

2.Requirements Description

2.1Introduction

The goal of this project is to build a catalog of software design patterns. The customer requires that the user can use this catalog to analyze different design patterns and choose the one that suits the most his needs for the given situation. The target clientele for this product are software developers who use Eclipse IDE for developing applications.

2.2General requirements (GR)

The product must be developed as an Eclipse plug-in which will be integrated with the ongoing project by the students in Paderborn who's goal is to create a pattern-oriented software development environment.
The product must enable organization and manipulation of design patterns. Manipulation is defined by the ability of the user to describe, manage, categorize, compare, relate and search for certain patterns.

2.3User requirements (UR)

User requirements define the tasks that a user should be able to do using the system.

UR01: The user should be able to manipulate multiple catalogs.
UR02: The user should be able to divide design patterns into multiple categories.
UR03: The user should be able to create, edit and delete patterns.
UR04: The user should be able to assign keywords to patterns.
UR05: The user should be able to search the patterns by key, full-text or categories.
UR06: The user should be able to compare the different patterns.
UR07: The user should be able to view and navigate through related patterns.
UR08: The user should be able to create, edit and delete pattern variants.

2.4Functional requirements (FR)

Functional requirements define what the system should be able to do.

FR01: The system should have an interface for listing, creating, editing and deleting multiple catalogs.
FR02: The system should have an interface for creating, editing and deleting categories of design patterns.
FR03: The system should have an interface for creating, editing and deleting of design patterns.
FR04: The system should be able to assign a design pattern to multiple categories.
FR05: The system should have an interface for adding, editing and deleting multiple descriptions of design patterns.
FR06: The system should be able to compare different design patterns.
FR07: The system should be able to categorize software patterns in a tree-like structure.
FR08: The system should be able to import a figure of the pattern structure.
FR09: The system should be able to import a figure describing the pattern relations.
FR10: The system should provide search by keyword, (sub)category, full-text search
FR11: The system should provide an interface for defining pattern keywords.
FR12: The system should provide navigation capabilities for navigating trough related patterns.
FR13: The system should have an interface for creating, editing and deleting design pattern variants.

2.5Nonfunctional requirements (NFR)

Nonfunctional requirements define standards that must be followed. They also define constraints on software design and implementation.

NFR01: Developing of the system in JAVA.
NFR02: Modeling done using EMF.
NFR03: Automated testing done with JUnit, EclEMMA.
NFR04: Final documentation done with LaTeX.
NFR05: The figures describing the relations of design patterns must be of jpg format.
NFR06: System developed as an Eclipse plug-in.
NFR07: Interfaces for integration with “Pattern-Oriented Software Engineering” project at Paderborn.
NFR08: The description types are:
1.Intent

2.Also known as
3.Motivation
4.Applicability
5.Structure
6.Participants
7.Collaborations
8.Consequences
9.Implementation
10.Sample code
11.Known uses
12.Related patterns
13.Other

NFR09: The relation types are:

1.Usable with
2.Alternative to
3.Excludes
4.Realized by
5.Other

3.Use Case Models

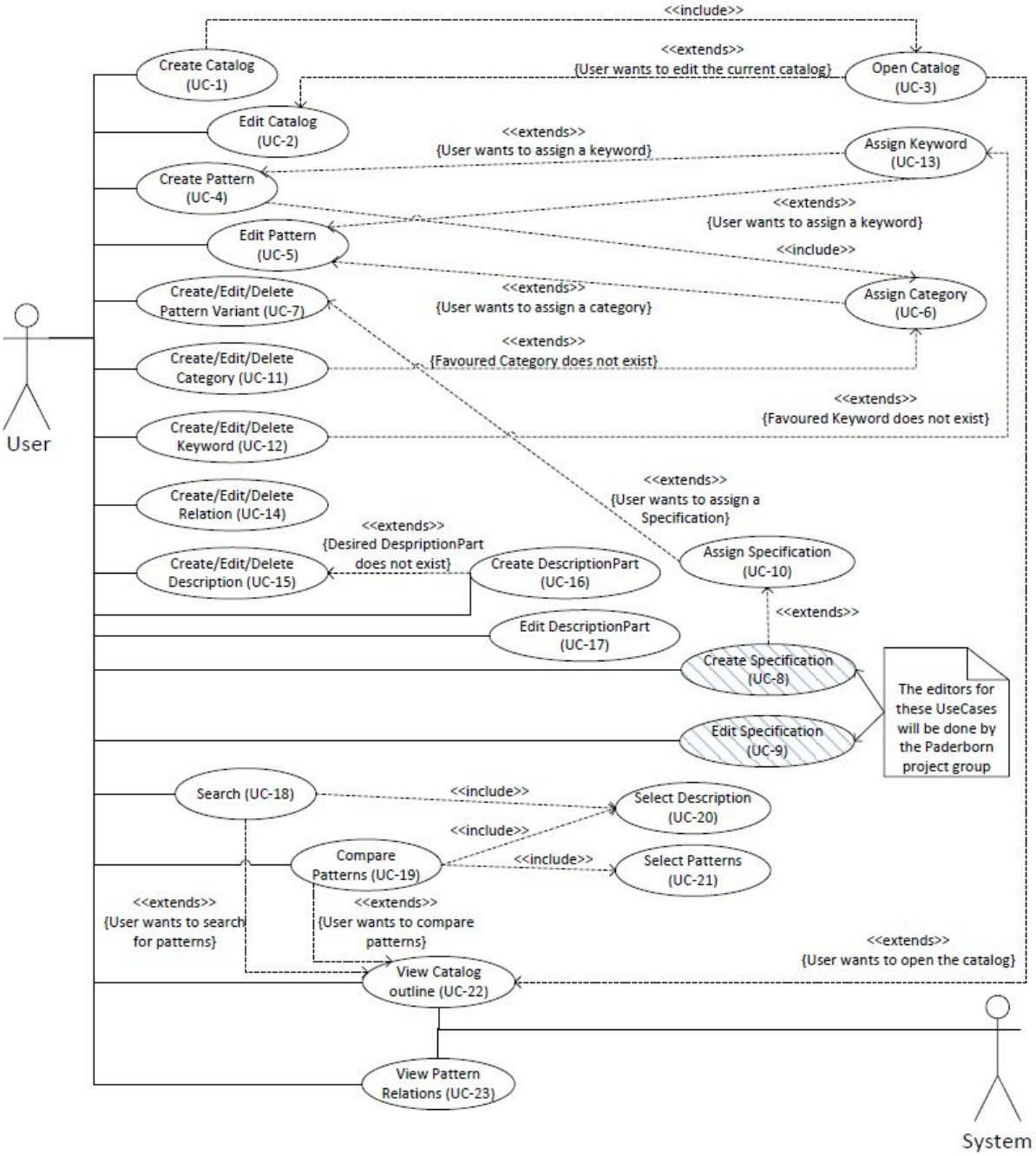

Illustration 1: Use case diagram

3.1Catalog administration

3.1.1Use case "Open pattern catalog"

Initiator:

User

Goal:

Open and view existing pattern catalog

Main Scenario:

1. On plug-in start up, a list of pattern catalogs is shown
2. User selects one pattern catalog and presses the 'Open' button
3. The system opens a 'Catalog view' with all the catalogs categories

Extensions:

User closes the main window by pressing on the 'Close' button

3.1.2Use case "Create pattern catalog"

Initiator:

User

Goal:

Create new pattern catalog

Main Scenario:

1. User opens the pattern catalog list (UC: Open pattern catalog)
2. User presses the 'Create new' button
3. The system creates a new catalog and shows an empty catalog form
4. User enters the required data (Catalog name)
5. User presses the 'Save' button
6. The system saves the catalog in a new file

Extensions:

User presses the 'Cancel' or 'Close' button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the catalog form.

3.1.3Use case "Edit pattern catalog"

Initiator:

User

Goal:

Edit pattern catalog

Main Scenario:

1. User opens the pattern catalog list (UC: Open pattern catalog)
2. User selects a catalog from the list and presses the 'Edit' button
3. The system shows a catalog form and enables the user to edit the catalog properties
4. User edits the catalog properties and presses the 'Save' button
5. The system saves all changes in the catalog file

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the catalog form.

3.1.4Use case “Delete pattern catalog”

Initiator:

User

Goal:

Delete pattern catalog

Main Scenario:

1. User opens the pattern catalog list (UC: Open pattern catalog)
2. User selects a catalog from the list and presses the ‘Delete’ button
3. The system prompts a warning message/question
4. User confirms the deletion
5. The system deletes the catalog file

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the catalog form.

3.1.5Use case “Open pattern”

Initiator:

User

Goal:

Open an existing pattern

Main Scenario:

1. User opens the catalog view (UC: Open pattern catalog) and selects a catalog
2. The system opens the pattern view and a list of all pattern categories is shown
3. User selects a category
4. The system opens all the patterns that belong to the selected category
5. User selects a pattern
6. The system opens a pattern form with all the pattern properties

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the catalog form.

3.1.6Use case “Create pattern”

Initiator:

User

Goal:

Create new pattern

Main Scenario:

1. User opens the catalog view (UC: Open pattern catalog) and selects a catalog

2. The system opens the pattern view and a list of all pattern categories is shown
3. User selects a category
4. The system opens all the patterns that belong to the selected category
5. User presses the ‘Create new’ button
6. The system creates a new pattern in the selected category and opens an empty pattern form
7. The user fills the required pattern fields and presses the ‘Save’ button
8. The system saves the pattern

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. The system shows a prompt message and after confirmation the system rejects all changes and closes the pattern form.

3.1.7Use case “Edit pattern”

Initiator:

User

Goal:

Edit an existing pattern

Main Scenario:

1. User opens the catalog view (UC: Open pattern catalog) and selects a catalog
2. User opens a pattern (UC: Open pattern)
3. User presses the ‘Edit’ button
4. The system shows an edit pattern form
5. User edits the pattern properties and presses the ‘Save’ button
6. The system saves all pattern changes

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. The system shows a prompt message and after confirmation the system rejects all changes and closes the pattern form.

3.1.8Use case “Delete pattern”

Initiator:

User

Goal:

Delete an existing pattern

Main Scenario:

1. User opens the catalog view (UC: Open pattern catalog) and selects a catalog
2. User opens the pattern view and selects a pattern category (UC: Open pattern)
3. User selects a pattern and presses the ‘Delete’ button
4. The system prompts a warning message/question
5. User confirms the deletion
6. User deletes the pattern and all its relations

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. The system shows a prompt message and after confirmation the system rejects all changes and closes the pattern form.

3.1.9Use case "Assign category"

Initiator:

User

Goal:

Assign a category to a pattern

Main Scenario:

1. User opens and edits a pattern (UC: Open pattern, Edit pattern)
2. User presses the 'Assign category' button
3. A list of categories and subcategories is shown in a hierarchical view
4. User selects a category and presses the 'Add' button
5. The system adds the category to the pattern category list
6. User presses the 'Save' button
7. The system saves changes to the pattern category list

Extensions:

User presses the 'Cancel' or 'Close' button. The system shows a prompt message and after confirmation the system rejects all changes and closes the Assign category form.

3.1.10Use case "Create pattern variant"

Initiator:

User

Goal:

Create new pattern variant

Main Scenario:

1. User opens the pattern catalog list (UC: Open pattern catalog)
2. User selects a pattern and opens the pattern view (UC: Open pattern)
3. User presses the 'Create new variant' button
4. The system creates a new variant for the opened pattern and shows the pattern variant in a new pattern view window
5. User enters the required fields and presses the 'Save' button
6. The system saves the pattern variant to the belonging pattern

Extensions:

User presses the 'Cancel' or 'Close' button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the pattern variant form.

3.1.11Use case "Edit pattern variant"

Initiator:

User

Goal:

Edit existing pattern variant

Main Scenario:

1. User opens a pattern (UC: Open pattern)
2. User enters the pattern edit mode (UC: Edit pattern)
3. User edits the pattern variant fields

4. User presses the ‘Save’ button
5. The system saves the pattern variant to the belonging pattern

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the pattern variant form.

3.1.12Use case “Delete pattern variant”

Initiator:

User

Goal:

Delete a pattern variant

Main Scenario:

1. User opens a pattern (UC: Open pattern)
2. User enters the pattern edit mode (UC: Edit pattern)
3. User selects a pattern variant and presses the ‘Delete’ button
4. The system prompts a warning message/question
5. User confirms deletion
6. The system deletes the pattern variant

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects any existing changes and closes the pattern editing form.

3.1.13Use case “Create pattern specification”

Initiator:

User

Goal:

Create pattern formal specification

Main Scenario:

1. User presses the ‘Create specification’ button
2. The system opens an external component for pattern specification creation
3. The user assigns the specification to a pattern (UC: Assign pattern specification)

3.1.14Use case “Edit pattern specification”

Initiator:

User

Goal:

Edit pattern formal specification

Main Scenario:

1. User opens a catalog (UC: Open catalog)
2. User opens a pattern (UC: Open pattern)
3. User presses the ‘Edit specification’ button
4. The system opens an external component for editing pattern specification

3.1.15Use case "Assign pattern specification"

Initiator:

User

Goal:

Assign pattern formal specification to a specific pattern

Main Scenario:

1. User opens a pattern (UC: Open pattern)
2. User presses the 'Assign specification' button
3. A list of all specifications is shown
4. User selects a specification and presses the 'Add' button
5. The system relates the selected specification to the open pattern

Extensions:

User presses the 'Cancel' or 'Close' button. A confirmation message is shown and after

3.1.16Use case "View categories"

Initiator:

User

Goal:

View catalog categories in a hierarchical view (tree view)

Main Scenario:

6. User opens a catalog (UC: Open catalog)
7. User presses the 'Edit categories' button
8. The system displays a list of categories in a hierarchical view

3.1.17Use case "Create category"

Initiator:

User

Goal:

Create a pattern category/subcategory

Main Scenario:

1. User opens the edit categories view (UC: View categories)
2. User selects a category or root element in the categories list
3. User enters a category name and presses the 'Add' button
4. The system adds the new category under the selected category/root

Extensions:

User presses the 'Cancel' or 'Close' button. A confirmation message is shown and after confirmation the system rejects all changes and closes the Edit categories view.

3.1.18Use case "Edit category"

Initiator:

User

Goal:

Edit a category name or hierarchical order

Main Scenario:

1. User opens the edit categories view (UC: View categories)
2. User selects a category
3. Using ‘Up’ and ‘Down’ buttons, the user moves the category up and down the hierarchical structure
4. User edits the category name and presses ‘Save’
5. The system saves the category’s new position and name

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the Edit categories view.

3.1.19Use case “Delete category”

Initiator:

User

Goal:

Delete a pattern category

Main Scenario:

1. User opens the edit categories view (UC: View categories)
2. User selects a category and presses the ‘Delete’ button
3. The system prompts a warning message/question
4. User confirms deletion
5. The system deletes the category and all its subcategories as well as all relations to their patterns

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the Edit categories view.

3.2Describing patterns

3.2.1Use case “Create keyword”

Initiator:

User

Goal:

Create a keyword for pattern organization

Main Scenario:

1. User opens a catalog (UC: Open catalog)
2. User selects the ‘View keywords’ item from the menu
3. A list of all keywords is shown as well as a form for entering a new one
4. User enters the keyword and presses the ‘Add’ button
5. The system checks for identical keywords and saves the new keyword

Extensions:

User presses the ‘Cancel’ or ‘Close’ button and the form is closed.
User is trying to create an existing keyword so the system displays a message and rejects saving of new keyword.

3.2.2Use case "Edit keyword"

Initiator:

User

Goal:

Edit an existing keyword

Main Scenario:

1. User opens a catalog (UC: Open catalog)
2. User selects the 'View keywords' item from the menu
3. A list of all keywords is shown as well as a form for entering a new one
4. User selects a keyword
5. User edits the keyword and presses the 'Save' button
6. The system checks for identical keywords and saves the new keyword

Extensions:

User presses the 'Cancel' or 'Close' button and the form is closed.
User is trying to create an existing keyword so the system displays a message and rejects saving of new keyword.

3.2.3Use case "Delete keyword"

Initiator:

User

Goal:

Delete an existing keyword

Main Scenario:

1. User opens a catalog (UC: Open catalog)
2. User selects the 'View keywords' item from the menu
3. A list of all keywords is shown as well as a form for entering a new one
4. User selects a keyword
5. User presses the 'Delete' button
6. The system prompts a message/warning
7. User confirms the deletion
8. The system deletes the keyword and all its relations with patterns

Extensions:

User presses the 'Cancel' or 'Close' button and the form is closed.

3.2.4Use case "Create pattern relation"

Initiator:

User

Goal:

Create a relation between two patterns

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the button 'Edit relations'
3. A list of all pattern's relations and relation types is shown
4. User presses the 'Create new relation' button

5. The system starts the create relations wizard
6. The system asks for a type of relation
7. User selects a type of relation
8. The system asks for the related pattern
9. User selects a related pattern and presses the ‘Save’ button
10. The system saves the new relation to both selected patterns.

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the create relations wizard.

3.2.5Use case “Edit pattern relation”

Initiator:

User

Goal:

Edit an existing relation between two patterns

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the button ‘Edit relations’
3. A list of all pattern’s relations and relation types is shown
4. User selects a relation and presses the ‘Edit’ button
5. The system enables editing of relation type
6. User selects a new relation type and presses the ‘Save’ button
7. The system saves the new relation type to the existing relation

Extensions:

User presses the ‘Cancel’ button. The system rejects all changes and disables editing of relation type.

3.2.6Use case “Delete pattern relation”

Initiator:

User

Goal:

Delete a pattern relation

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the button ‘Edit relations’
3. A list of all pattern’s relations and relation types is shown
4. User selects a relation an presses the ‘Delete’ button
5. The system prompts a message/warning
6. The user confirms deletion
7. The system deletes the selected relation

Extensions:

User presses ‘Cancel’ button. The system rejects the deletion and returns to edit relations view.

3.2.7Use case “Create description”

Initiator:

User

Goal:

Create a pattern description

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the ‘Add description’ button
3. The system starts the add description wizard
4. The system asks for the description type and title
5. User selects description type and enters the title
6. The system creates the first Description part
7. The user edits the description part (UC: Edit description) and presses the ‘Save’ button
8. The system saves the description and description part to the pattern

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the create description wizard.

3.2.8Use case “Delete description”

Initiator:

User

Goal:

Delete pattern description and all its parts

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the ‘Edit description’ button next to the description he wants to edit
3. The system shows the description and description parts in an editor
4. User presses the ‘Delete description’ button
5. The system prompts a message/warning
6. User confirms deletion
7. The system deletes the description and its parts

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the description editor.

3.2.9Use case “Create description part”

Initiator:

User

Goal:

Creating a description part

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the ‘Edit description’ button next to the description he wants to edit
3. The system shows the description and description parts in an editor
4. User presses the ‘Add part’ button
5. The system asks for the part type (text or figure)
6. User selects part type
7. The system opens the part editor according to selected type
8. User enters the part description or figure and presses the ‘Save’ button
9. The system saves the part to the selected pattern description

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the description editor.

3.2.10Use case “Edit description part”

Initiator:

User

Goal:

Editing a description part

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the ‘Edit description’ button next to the description he wants to edit
3. The system shows the description and description parts in an editor
4. User edits an existing part and presses the ‘Save’ button
5. The system saves all changes and disables the part editor

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the description editor.

3.2.11Use case “Delete description part”

Initiator:

User

Goal:

Delete description part

Main Scenario:

1. User opens the edit pattern view (UC: Edit pattern)
2. User presses the ‘Edit description’ button next to the description he wants to edit
3. The system shows the description and description parts in an editor
4. User presses the ‘Delete part’ button next to the part he wants to delete
5. The system prompts a message/warning
6. User confirms deletion
7. The system deletes the selected part from the pattern description

Extensions:

User presses the ‘Cancel’ or ‘Close’ button. A confirmation message is shown and after confirmation the system rejects all changes and closes the description editor.

3.3Exploring patterns

3.3.1Use case “Search patterns”

Initiator:

User

Goal:

Search and find desired pattern

Main Scenario:

1. User selects and opens a catalog (UC: Open catalog)

2. User selects the ‘Search’ item from the menu
3. The system opens the search engine
4. A list of search options is displayed (full text search, details search, specification search, keyword search…)
5. User selects the type of search which he wants to use
6. User enters the search keywords and presses the ‘Search’ button
7. The system searches the selected patterns and displays the result as a list of pattern titles (with links to patterns) with the pattern options in which the results are found

Extensions:

User presses the ‘Close’ button. The search window is closed.

3.3.2Use case “Compare patterns”

Initiator:

User

Goal:

Comparison of selected patterns

Main Scenario:

1. User selects and opens a catalog (UC: Open catalog)
2. User selects the ‘Compare patterns’ item from the menu
3. The system opens the search engine and next to it an empty pattern comparison list
4. User finds a pattern using the search (UC: Search patterns)
5. User presses the ‘Add pattern to comparison list’ button
6. The system adds the selected pattern to the comparison list
7. After filling the list, the user selects the ‘Compare selected patterns’ button
8. The system opens the compare patterns wizard
9. The system asks the user to order his patterns with ‘Up’ and ‘Down’ buttons
10. User orders the patterns and presses the ‘Next’ button
11. The system asks the user to select details and detail parts he wants to compare
12. User selects details and details parts
13. The system stacks the patterns vertically from left to right aligning the selected details and details parts

Extensions:

User clicks the ‘Close’ button. The system closes the compare patterns view.

3.3.3Use case “View pattern relations”

Initiator:

User

Goal:

Viewing pattern relations

Main Scenario:

1. User selects and opens a catalog (UC: Open catalog)
2. User selects the ‘View patterns’ item from the menu
3. The system opens the search engine and next to it an empty pattern list
4. User finds a pattern using the search (UC: Search patterns)
5. User presses the ‘Add pattern relations list’ button
6. The system adds the selected pattern to the comparison list
7. After filling the list, the user selects the ‘View relations’ button
8. The system generates and opens a graphical view of the selected patterns relations

Extensions:

User clicks the ‘Close’ button. The system closes the pattern relations view.

3.4Integration with the tool from Paderborn

3.4.1Use case “Open pattern specification editor”

Initiator:

User

Goal:

Open pattern specification editor and edit pattern specification

Main Scenario:

1. User opens the edit pattern view (UC: edit pattern)
2. User presses the ‘Open pattern specification editor’ item from the menu
3. The system opens the external pattern specification editor

3.4.2Use case “Start pattern application”

Initiator:

User

Goal:

Start pattern application

Main Scenario:

1. User selects and opens a pattern (UC: open pattern)
2. User selects the ‘Start pattern application’ item from the menu
3. The external pattern application for the selected pattern is started

Extensions:

The selected pattern doesn’t have a specification. The system prompts a message

User selects ‘Close’ button. The system closes the pattern view.

4.Requirements Definition

4.1Requirement Group Definitions

Identification	Requirement Group	Rem.
CA	Catalog Administration	
DP	Describing Patterns	
EP	Exploring Patterns	
IP	Integration with the tool from Paderborn	

4.2Requirement Sources

Source	Description	Rem.
DT	Dietrich Travkin	
MD	Markus von Detten	

4.3Requirements definitions

Identity	Status	Priority	Description	Source
			Catalog Administration	
CA-1	I	1	Open pattern catalog	DT, MD
CA-2	I	1	Create pattern catalog	DT, MD
CA-3	I	1	Edit pattern catalog	DT, MD
CA-4	I	1	Delete pattern catalog	DT, MD
CA-5	I	1	Open pattern	DT, MD
CA-6	I	1	Create pattern	DT, MD
CA-7	I	1	Edit pattern	DT, MD
CA-8	I	1	Delete pattern	DT, MD
CA-9	I	1	Assign cattegory	DT, MD
CA-10	I	1	Create pattern variant	DT, MD
CA-11	I	1	Edit pattern variant	DT, MD
CA-12	I	1	Delete pattern variant	DT, MD
CA-13	I	1	Create pattern specification	DT, MD
CA-14	I	1	Edit pattern specification	DT, MD
CA-15	I	1	Assign pattern specification	DT, MD
CA-16	I	1	View categories	DT, MD
CA-17	I	1	Create category	DT, MD
CA-18	I	1	Edit category	DT, MD
CA-19	I	1	Delete category	DT, MD
			Describing Patterns	
DP-1	I	1	Create keyword	DT, MD
DP-2	I	1	Edit keyword	DT, MD
DP-3	I	1	Delete keyword	DT, MD
DP-4	I	1	Create pattern relation	DT, MD
DP-5	I	1	Edit pattern relation	DT, MD
DP-6	I	1	Delete pattern relation	DT, MD
DP-7	I	1	Create description	DT, MD
DP-8	I	1	Delete description	DT, MD
DP-9	I	1	Create description part	DT, MD
DP-10	I	1	Edit description part	DT, MD
DP-11	I	1	Delete description part	DT, MD
			Exploring Patterns	
EP-1	I	2	Search patterns	DT, MD
EP-2	I	2	Compare patterns	DT, MD
EP-3	I	2	View pattern relations	DT, MD
			Integration with the tool from Paderborn	
IP-1	I	2	Open pattern specification editor	DT, MD
IP-2	I	2	Start pattern application	DT, MD

Requirement status:

I = initial (this requirement has been identified at the beginning of the project),
D = dropped (this requirement has been deleted from the requirement definitions),
H = on hold (decision to be implemented or dropped will be made later),
A = additional (this requirement was introduced during the project course).

4.3.1Change Log

Identity	Action	Date	Comments

Requirement status:
D = dropped (this requirement has been deleted from the requirement definitions),
H = on hold (decision to be implemented or dropped will be made later),
A = added (this requirement was introduced during the project course).
R = resurrected (dropped or on hold requirement was reactivated)

5.Future Development

5.1General Overview

Future development could consist of making editors for defining pattern structure and pattern relations as part of the plug-in instead of importing figures that describe the pattern structure.