



SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

PUML : Project UML

**Version [1.1]
[12/16/2011]**

**Prepared for:
Bruce Bolden, Dr. Clinton Jeffery**

**Prepared by:
Coleman Beasley, Alex Dean, Austin Enfield, Jason Fletcher, Theora Rice, Cable Johnson,
Adrian Norris, David Summers**

**Emeritus Contributors:
Noel Klein, Max McKinnon**

**University of Idaho
Moscow, ID 83844-1010**

**PUML SSRS
TABLE OF CONTENTS**

Section	Page
1 Introduction (Alex)	1
1.1 IDENTIFICATION	1
1.2 PURPOSE	1
1.3 SCOPE	1
1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	1
1.5 REFERENCES	1
1.6 OVERVIEW AND RESTRICTIONS	1
2 OVERALL DESCRIPTION (CKB)	2
2.1 PRODUCT PERSPECTIVE	2
2.2 PRODUCT FUNCTIONS	2
2.3 USER CHARACTERISTICS	2
2.4 CONSTRAINTS	2
2.5 ASSUMPTIONS AND DEPENDENCIES	2
2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS	2
2.6.1 Site dependencies	2
2.6.2 Safety, security and privacy requirements	3
2.6.3 Performance requirements	3
2.6.4 System and software quality	3
2.6.5 Packaging and delivery requirements	3
2.6.6 Personnel-related requirements	3
2.6.7 Training-related requirements	3
2.6.8 Logistics-related requirements	3
2.6.9 Other requirements	3
2.6.10 Precedence and criticality of requirements	3
3 SPECIFIC REQUIREMENTS (Noel)	4
3.1 EXTERNAL INTERFACE REQUIREMENTS	4
3.1.1 Hardware Interfaces	4
3.1.2 Software Interfaces	4
3.1.3 User Interfaces	4
3.1.4 Other Communication Interfaces	4
3.2 SYSTEM FEATURES	1
3.2.1 Use Case Diagrams	1
3.2.2 Use Case Descriptions	2
3.2.2.1 Create new diagram - Req No. UC1	2
3.2.2.2 Load a diagram - Req No. UC2	2
3.2.2.3 Export a diagram - Req No. UC3	3

3.2.2.4	Save a diagram - Req No. UC4	3
3.2.2.5	Print a diagram - Req No. UC5	3
3.2.2.6	Toggle grid on/off - Req No. UC6	4
3.2.2.7	Place shapes on diagram - Req No. UC7	4
3.2.2.8	Connect shapes - Req No. UC8	4
3.2.2.9	Undo commands - Req No. UC9	5
3.2.2.10	Redo commands - Req No. UC10	5
3.2.2.11	Select shapes and multiple shapes - Req No. UC11	5
3.2.2.12	Move shapes - Req No. UC12	5
3.2.2.13	Copy shapes - Req No. UC13	6
3.2.2.14	Cut shapes - Req No. UC14	6
3.2.2.15	Paste shapes - Req No. UC15	6
3.2.2.16	Edit shape properties - Req No. UC16	6
3.2.2.17	Edit default properties - Req No. UC17	7
4	REQUIREMENTS TRACEABILITY (Noel and Max)	1
5	APPENDIX A. List of Requirements	1

1 Introduction (Alex)

1.1 IDENTIFICATION

The software system being considered for development is referred to as pUML. The customer providing specifications for the system is Bruce Bolden. The ultimate customer, or end-user, of the system will be Bruce Bolden and his associates. This is a new project effort, so the version under development is version 1.0.

1.2 PURPOSE

The purpose of the system under development is to develop a flexible and easy to use UML diagram editor. While the system will be used by Bruce Bolden and his associates, this document is intended to be read and understood by UICS software designers and coders.

1.3 SCOPE

The system being developed was started on October 12th, 2011. The project sponsor and user is Bruce Bolden. The developers are Coleman Beasley, Alex Dean, Jason Fletcher, Noel Klein, Max McKinnon, and Theora Rice.

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Term or Acronym	Definition
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification

1.5 REFERENCES

Qt - drag and drop with graphics view framework Internet. Available: <http://stackoverflow.com/questions/2732343/qt-drag-and-drop-with-graphics-view-framework>

1.6 OVERVIEW AND RESTRICTIONS

This document is for limited release only to UI CS personnel working on the project.

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.

Sections 5 and up are appendices including original information and communications used to create this document.

2 OVERALL DESCRIPTION (Coleman)

2.1 PRODUCT PERSPECTIVE

This software is completely independent and self-contained.

2.2 PRODUCT FUNCTIONS

Our product is a Unified Modeling Language (UML) diagram editor. It creates 5 different types of diagrams and exports them into user-friendly formats. The editor streamlines the creation process by making available only shapes that apply to the current diagram. Multiple shape and line colors are supported to allow for more versatile diagrams.

2.3 USER CHARACTERISTICS

The users of the product will be software developers who are in the planning phases of their project. Users are expected to be familiar with UML diagrams and understand the basic idea behind each drawing. The software will attempt to make the process easier, but a basic understanding will be necessary to use the editor effectively.

2.4 CONSTRAINTS

There are no applicable constraints.

2.5 ASSUMPTIONS AND DEPENDENCIES

The software is built with the assumption that the hardware and operating system it is installed on will support the Qt graphics library.

2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

2.6.1 Site dependencies

There are no anticipated site dependencies.

2.6.2 Safety, security and privacy requirements

There are no safety, security or privacy requirements for the system.

2.6.3 Performance requirements

The program is intended for use by one user at a time. There are no other performance requirements.

2.6.4 System and software quality

The software shall create, edit and save 5 types of UML diagrams. It is designed to operate in the same general manner for each diagram. The software is designed as modularly as possible to allow for ease of modification. It is designed to be run on as many systems as possible. For this reason, it is being developed in C++ using an open source graphics system.

2.6.5 Packaging and delivery requirements

The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD's and/or via email, as specified by the customer at time of delivery. Although document "drops" will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.

2.6.6 Personnel-related requirements

The system under development has no special personnel-related requirements.

2.6.7 Training-related requirements

No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

2.6.8 Logistics-related requirements

The system requires a computer that can run Windows, Mac OSX, or Linux (x86 or x64). The only peripheral hardware needs are a mouse, keyboard, monitor, and printer(optional)

2.6.9 Other requirements

There are no other requirements.

2.6.10 Precedence and criticality of requirements

No requirements have been defined as critical.

3 SPECIFIC REQUIREMENTS (Noel)

This section of the document contains all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement is externally perceivable by users, operators, or other external systems. All requirements include at a minimum a description of every input into the system, every output from the system, and all functions performed by the system in response to an input or in support of an output.

3.1 EXTERNAL INTERFACE REQUIREMENTS

This subsection is a detailed description of all inputs into and outputs from the software system. It complements the constraints and dependencies defined in earlier sections. Hardware, software, user, and other communication interfaces are specified.

3.1.1 Hardware Interfaces

For the purpose of the UML application, there is no special hardware necessary. Only standard computer interfaces, such as a monitor, a keyboard and a mouse are required. A printer is necessary, if the user wishes to print out the diagrams.

3.1.2 Software Interfaces

The customer did not demand any specific software interfaces. There is also no special interface for the development and function necessary. Only the print dialog will connect to the users printer driver software or use the OS's default dialog.

3.1.3 User Interfaces

The user will be able to use the usual computer input types, such as keyboard and mouse to use the graphical user interface of the UML tool.

3.1.4 Other Communication Interfaces

No other communication interfaces are needed.

External Interface Requirements

Hardware Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
Keyboard	Keyboard/Text fields	Text input	Standard		
Mouse	Mouse/GUI	For using the GUI	Standard		
Monitor	GUI/Monitor	Display GUI	Standard		
Printer	Diagram/Printer	Print diagram	Standard		

Software Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
Print dialog	OS/UML tool	Standard OS printer dialog			

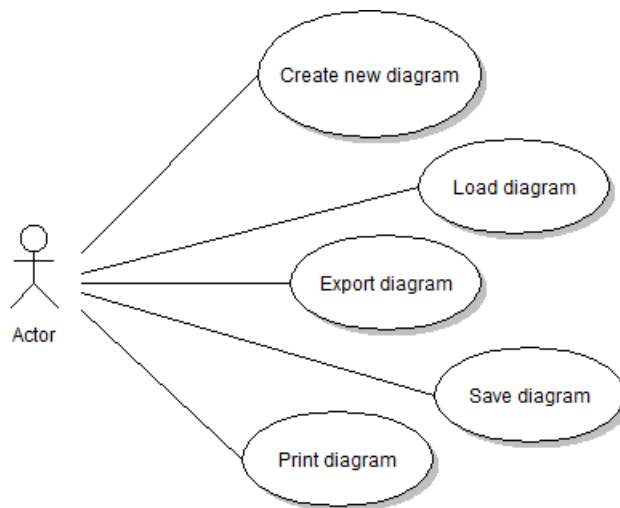
User Interfaces

Name	Source/Destination	Description	Type/range	Dependencies	Formats
GUI	GUI/Monitor	The UML tool is used via the GUI			

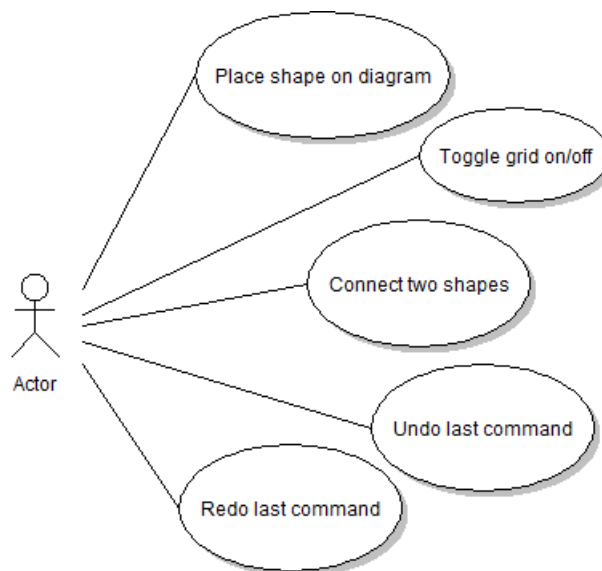
3.2 SYSTEM FEATURES

The functional requirements in this section define the fundamental actions (i.e., features) that must take place in the UML tool in accepting and processing the inputs and in processing and generating the outputs. These requirements are given in the form of **Use Cases** where possible, denoting a concrete use (discrete user-performable task) of the system. Use case diagrams are followed by use case descriptions.

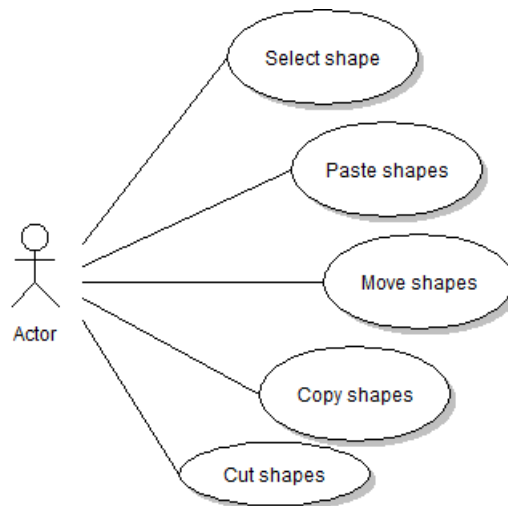
3.2.1 Use Case Diagrams



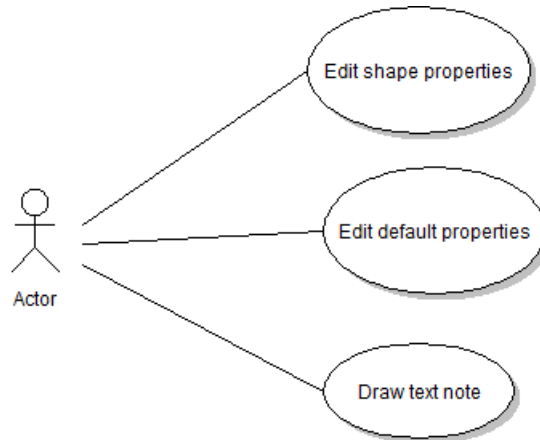
File operations



Drawing use cases 1



Drawing use cases 2



Shape properties

3.2.2 Use Case Descriptions

3.2.2.1 Create new diagram - Req No. UC1

Actors - User Goals - Create a new diagram

Preconditions - UML editor is started

Summary - The user creates a new diagram.

Related use cases - none

Steps

1. Click new on the top toolbar.
2. A dialogue pops up. Choose the diagram type in the dialogue.
3. Click create.

Alternatives - none

Post conditions - There is an empty diagram. The diagram specific tools are activated on the left toolbar. One can now start drawing a UML diagram.

3.2.2.2 Load a diagram - Req No. UC2

Actors- User

Goals- Load a previously saved diagram

Preconditions - UML editor is started

Summary - The user loads a previously saved diagram from the file system.

Related use cases - none

Steps

1. Click open on the top toolbar.
2. A dialog pops up. Navigate through your file system. Choose the file.
3. Click open.

Alternatives - none

Post conditions - The saved diagram is open. The diagram specific tools are activated on the left toolbar.

One can now start editing the UML diagram.

3.2.2.3 Export a diagram - Req No. UC3

Actors - User

Goals - Export the open diagram

Preconditions - Diagram file is opened

Summary - The user exports the open diagram to a PDF file.

Related use cases - none

Steps

1. Click save on the top toolbar.
2. Click export to PDF in the pop-down menu.
3. Enter a filename.
4. Click save.

Alternatives - none

Post conditions - The diagram is exported as a PDF file. The PDF file is saved in the user specified location of the users file system.

3.2.2.4 Save a diagram - Req No. UC4

Actors - User

Goals - Save the open diagram

Preconditions - Diagram file is opened

Summary - The user saves the open diagram to a XML file.

Related use cases - none

Steps

1. Click save on the top toolbar.
2. Click save as file in the pop-down menu.
3. Enter a filename.
4. Click save.

Alternatives - none

Post conditions - The diagram is saved as an XML file. The XML file is saved in the user specified location of the users file system.

3.2.2.5 Print a diagram - Req No. UC5

Actors - User

Goals - Print the open diagram

Preconditions - Diagram file is opened

Summary - The user prints the open diagram. The UML tool uses the OS's print dialog and therefore the OS's print drivers.

Related use cases - none

Steps

1. Click print on the top toolbar.
2. Select a printer.
3. Select printer specific options.
4. When you are satisfied with the settings, click print.

Alternatives - none

Post conditions - The diagram is printed.

3.2.2.6 Toggle grid on/off - Req No. UC6

Actors - User

Goals - Toggle the grid of the drawing area on or off

Preconditions - Diagram file is opened

Summary - The user toggles the grid of the drawing area on or off.

Related use cases - none

Steps

1. Click on grid on the top toolbar.

Alternatives - none

Post conditions - If the grid was turned on, the grid is turned off. If the grid was turned off, the grid is turned on.

3.2.2.7 Place shapes on diagram - Req No. UC7

Actors - User

Goals - Place a shape on the drawing area

Preconditions - Diagram file is opened

Summary - The user chooses a shape from the drawing toolbar and places it on the drawing area.

Related use cases - none

Steps

1. Select the shape from the left drawing toolbar.
2. Click on the diagram to place the shape on the diagram.

Alternatives - none

Post conditions - The selected shape is placed on the drawing area at the place pointed at by the user.

3.2.2.8 Connect shapes - Req No. UC8

Actors - User

Goals - Connect two shapes on the drawing area with a connecting line

Preconditions - Diagram file is opened

Summary - The user chooses a line type from the drawing toolbar and specifies the two shapes on the drawing area to connect.

Related use cases - none

Steps

1. Select a connector from the left drawing toolbar.
2. Click on the source shape.
3. Click on the target shape.
4. A connecting line is placed from the source shape to the target shape.

Alternatives - none

Post conditions - The selected line connects is placed on the drawing area at the place pointed at by the user.

3.2.2.9 Undo commands - Req No. UC9

Actors - User

Goals - Undo the last command

Preconditions - Diagram file is opened. A command has been used.

Summary - The last action taken by the user is being undone and put into a redo stack.

Related use cases - none

Steps

1. Click undo on the top toolbar.
2. The last command is revoked.
3. Continue drawing.

Alternatives - none

Post conditions - The last command is revoked and stored in the redo.

3.2.2.10 Redo commands - Req No. UC10

Actors - User

Goals - Redo the last undone command

Preconditions - Diagram file is opened. A command has been undone.

Summary - The last action on top of the redo stack is redone and removed from the stack.

Related use cases - none

Steps

1. Click redo on the top toolbar.
2. The last revoked command is redone.
3. Continue drawing.

Alternatives - none

Post conditions - The last undone action is redone and removed from the stack.

3.2.2.11 Select shapes and multiple shapes - Req No. UC11

Actors - User

Goals - Select one or more shapes

Preconditions - Shapes are available to select on the drawing area.

Summary - The user selects the shapes for further commands.

Related use cases - none

Steps// 1. Click on a shape to select it.

2. Hold CTRL and click other shapes to select multiple shapes.

Alternatives - none

Post conditions - The selected shapes are marked. Further commands can be invoked to all selected shapes.

3.2.2.12 Move shapes - Req No. UC12

Actors - User

Goals - Move one or more shapes

Preconditions - One or more shapes are selected

Summary - The user moves the shapes to a different location on the drawing area.

Related use cases - none

Steps

1. Select one or more shapes.
2. Click and hold the mouse button on one of the shapes.
3. Move the shapes to a new place.
4. Release the mouse button

Alternatives - none

Post conditions - The selected shapes are moved to the new place.

3.2.2.13 Copy shapes - Req No. UC13

Actors - User

Goals - Copy one or more shapes

Preconditions - One or more shapes are selected

Summary - The user copies the selected shapes to the clipboard.

Related use cases - none

Steps

1. Select one or more shapes.
2. Click copy on the top toolbar.

Alternatives - none

Post conditions - The selected shapes are copied to the clipboard and can be pasted in a drawing area.

3.2.2.14 Cut shapes - Req No. UC14

Actors - User

Goals - Cut one or more shapes

Preconditions - One or more shapes are selected

Summary - The user cuts the selected shapes to the clipboard. The shapes are removed from the drawing area.

Related use cases - none

Steps

1. Select one or more shapes.
2. Click cut on the top toolbar.

Alternatives - none

Post conditions - The selected shapes are copied to the clipboard and can be pasted in a drawing area. The shapes are removed from the drawing area.

3.2.2.15 Paste shapes - Req No. UC15

Actors - User

Goals - Paste one or more shapes

Preconditions - A selection of shapes is stored in the clipboard

Summary - The user pastes the shapes from the clipboard into the drawing area.

Related use cases - none

Steps

1. Click paste on the top toolbar.

Alternatives - none

Post conditions - The shapes from the clipboard are inserted in the top left corner of the diagram.

3.2.2.16 Edit shape properties - Req No. UC16

Actors - User

Goals - Change the properties of a single shape

Preconditions - A shape is available for editing on the draw area.

Summary - The user edits a shape on the drawing area.

Related use cases - none

Steps

1. Select one shape.
2. Click shape preferences on the top toolbar.

3. Select shape specific options.

4. When you are satisfied with the settings, click ok.

Alternatives - Instead of selecting a shape and clicking shape preferences on the toolbar, one can also double click on a shape to get to the shape specific options.

Post conditions - The shape properties are changed.

3.2.2.17 Edit default properties - Req No. UC17

Actors - User

Goals - Change the default properties of the UML tool

Preconditions - UML editor is started

Summary - The user edits the default properties of the UML tool.

Related use cases - none

Steps

1. Click on options on the top toolbar.

2. Modify preferences for the default shapes.

The shape options are:

- Color
- Fill color
- Line thickness

The line options are:

- Color
- Thickness

The text options are:

- Color
- Font
- Size
- Fill color
- Border color

Alternatives - none

Post conditions - The default properties are changed.

4 REQUIREMENTS TRACEABILITY (Noel)

Feature Name	Req No.	Requirement Description	Priority
Create a new diagram	UC1	The user creates a new diagram.	Mandatory
Load a diagram	UC2	The user loads a previously saved diagram from the file system.	Mandatory
Export a diagram	UC3	The user exports the open diagram to a PDF file. Related use cases	low
Save a diagram	UC4	The user saves the open diagram to a XML file. Related use cases	Mandatory
Print a diagram	UC5	The user prints the open diagram. The UML tool uses the OS's print dialog and therefore the OS's print drivers.	low
Toggle grid on/off	UC6	The user toggles the grid of the drawing area on or off.	high
Place shapes on diagram	UC7	The user chooses a shape from the drawing toolbar and places it on the drawing area.	Mandatory
Connect shapes	UC8	The user chooses a line type from the drawing toolbar and specifies the two shapes on the drawing area to connect.	Mandatory
Undo commands	UC9	The last action taken by the user is being undone and put into a redo stack.	low
Redo commands	UC10	The last action on top of the redo stack is redone and removed from the stack.	low
Select shapes	UC11	The user selects the shapes for further commands.	low
Move shapes	UC12	The user moves the shapes to a different location on the drawing area.	high
Copy shapes	UC13	The user copies the selected shapes to the clipboard.	low
Cut shapes	UC14	The user cuts the selected shapes to the clipboard. The shapes are removed from the drawing area.	low
Paste shapes	UC15	The user pastes the shapes from the clipboard into the drawing area.	low
Edit shape properties	UC16	The user edits a shape on the drawing area.	low
Edit default properties	UC17	The user edits the default properties of the UML tool.	low

Priorities are: **Mandatory**, **Low**, **High**

5 APPENDIX A. List of Requirements

- Should support the five main types of diagrams
Should have a library of UML Graphical Objects
- File (Open/Close/Save/Import)
- Easy to use
- Portable-ish
- Light on System Resources
- Cut/Copy/Duplicate/Paste
- Undo/Redo
- Keyboard Bindings
- Template Generation
- Diagram from code
- UML syntax parser
- Print and export to graphics, cover page generation
- GUI
- Support for defects
- Installer
- Collaborative Support
- Project Manager (File Organizer)/File Linking
- Multiple Documents
- Man page/Help/user manual/tutorials
- Color
- Snap to grid/alignment
- Scaling/zoom
- Font selection
- Spell check
- Change settings
- Default UML Diagram template
- Text editor (LaTeX)
- Tool Tips, in many forms

- Collapse branches, trees
- Integrate with other UML editors
- File Format (internal)
- XML MUST FORMAT
- Script to auto-save and create backups
- Special character support
- FIND and replace