Fontys University of Applied Sciences

User Requirements Specification Document for Pipelines

Coen Stange, Edgar Kruze, Wen Li

2015/11/25

Background and Context

This User Requirements Specification document specifies the requirements for the application "Pipelines in a flow network". To transport fuel from one place to another place pipelines are used. The system is becoming more complex over time therefore it becomes difficult to manage the flow of in the pipeline system. For safety purposes it is very important that the pressure in the pipes don't become to high. The following components are part of the flow system: pump, sink, adjustable splitter and merger.

Contents

1 Functional Requirements

1.1 Requirements

1.1.1 General requirements

Code	Requirement
GEN-010	The program is compatible with Windows 7.
GEN-020	The system allows to design a flow system using the folowing components: • Pump • Sink • Adjustable splitter • Merger
GEN-030	The system shows the flow through the pipelines.

1.1.2 System requirements

Code	Requirement
SYS-010	The system allows to add components to the flow network.
SYS-020	The components are not allowed to overlap each other.
SYS-030	The system allows to connect a pipeline between an unused output of a component to an unused input of a component.
SYS-040	The system allows to remove components including pipelines.
SYS-040A	All the pipelines connected to the removed component will also be removed.
SYS-050	The system allows to change the flow output of a pump within his range.
SYS-060	The system allows to change the distribution of the flow from splitter going to the top and bottom output.
SYS-070	The system displays the flow going through the pipelines.
SYS-070A	The system displays the pipeline red in case the flow exceeds the safety limit.
SYS-080	The system displays the flow transported to a sink.
SYS-090	The system allows to export the flow network to a file.
SYS-100	The system allows to import a flow network from a file.

1.2 Use cases

Use Case 1	**Adding new component on flow network**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Component selected from toolbar

Main Success Scenario:

1. User selects place on flow network grid
2. System places component on grid
3. System updates internal state

Extensions:

2.a System can't place component in selected area

1. System informs end-user that component can't be placed, because on that point already exists component.
2. End of use case

Use Case 2	**Changing flow output/input in pump/tank**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Selected pump/tank on flow network grid

Main Success Scenario:

1. User selects component on grid
2. System opens property sidebar
3. User input new output/input values in properties sidebar
4. System updates values of component
5. System updates internal state

Extensions:

2.a System can't open property bar, because user selected wrong component

1. User returns to step 1.
2. End of use case

Use Case 3	**Changing flow output in splitter component**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Selected pump/tank on flow network grid

Main Success Scenario:

1. User selects component on grid
2. System opens property sidebar
3. User input new output values in properties sidebar
4. System updates values of component
5. System updates internal state

Extensions:

2.a System can't open property bar, because user selected wrong component

1. User returns to step 1.
2. End of use case

Use Case 4	**Creating connection between components**

Level:	User-goal
Primary Actor:	End-User
Preconditions:	Selected pipeline drawing tool from toolbar

Main Success Scenario:

1. User selects starting component
2. User selects point on grid, through where pipeline will go
3. User repeats step 2, untill he has selected all points, through where pipeline will go
4. User selects ending component
5. Systems draws pipeline
6. System updates internal state

Extensions:

1.a Selected component already has no unused output

   1. System informs end-user that the selected component can't be used.
   2. User returns to step 1.
   3. End of use case.

2.a Selected point on grid is occupied by component.

   1. System ignores the selected point.

Use Case 5	**Deleting element on flow network grid**
Level:	User-goal

Primary Actor:	End-User
Preconditions:	Selected tool for removing component

Main Success Scenario:

1. User selects component on pipeline network grid
2. System removes selected component
3. System removes pipelines connected to component, if component isn't pipeline
4. System updates internal state

Use Case 6	**Changing current flow of a pump**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Selected pump

Main Success Scenario:

1. User inputs new flow
2. System updates flow network grid.

Extensions:

1.a The new flow exceeds capacity

    1. System informs end-user that the selected component can't be used and the maximum flow possible is used

Use Case 7	**Change output adjustable splitter**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Selected adjustable splitter

Main Success Scenario:

1. User uses slider to adjust the output.
2. System updated the flow network grid.

Extensions:

1.a Component isn't adjustable splitter.

1. The option to adjust the flow isn't shown.

Use Case 8	**Current flow**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Pipeline connection

Main Success Scenario:

1. System displays current flow in pipeline.

Use Case 9	**Load file**
Level:	User-goal

Primary Actor:	End-User
Preconditions:	Saved file with pipeline network

Main Success Scenario:

1. User selects file to open.
2. System closes previous project.
3. System loads the file.

Extensions:

2.a File can't be loaded.

1. System informs user that file can't be loaded and given the choice to stop or choose another file.
2. End of use case.

2.b Another project is open

1. System asks if users wants, to save project, that is already open, before closing it and opening another project.
2. End of use case.

Use Case 10	**Save file**
Level:	User-goal
Primary Actor:	End-User
Preconditions:	Open pipeline network project

Main Success Scenario:

1. User presses save button.
2. System saves the file.

---

2 | User Interface

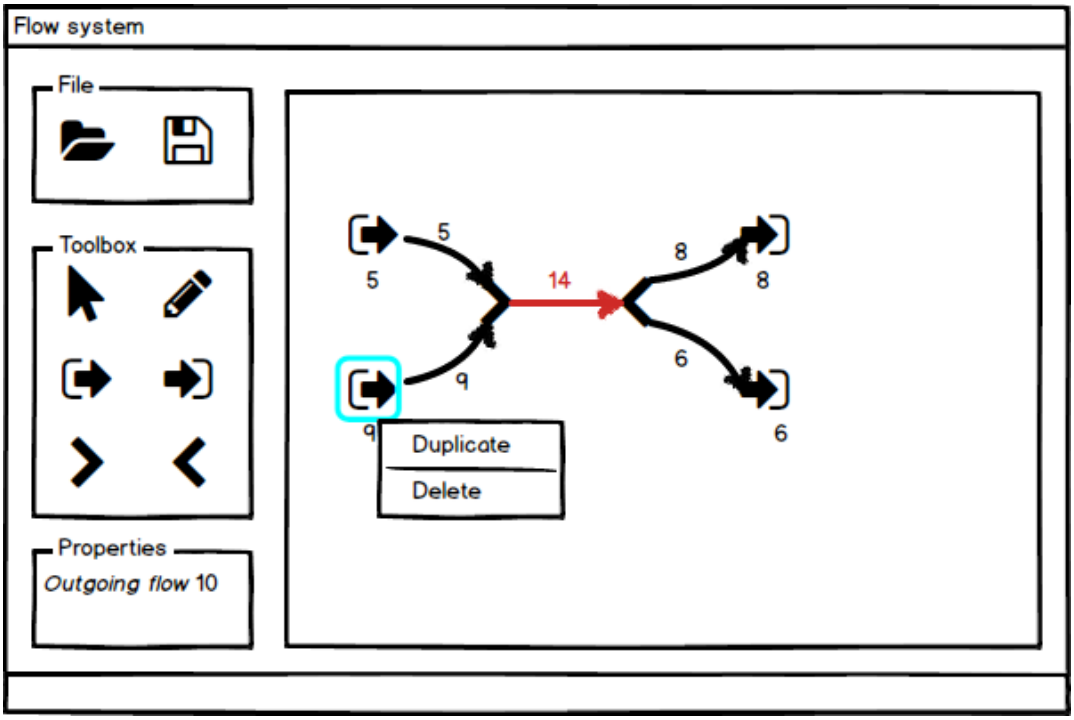

Figure 2.1: Mockup

2.1 Specifications

Code	Specification
UIS-010	An open file dialog box will be showed by pressing .
UIS-020	A save file dialog box will be showed by pressing .
UIS-030	The sidebar contains a toolbox.
UIS-030A	The application has different modes, only one button can be selected in the toolbox.
UIS-040	The normal mode is selected by default .
UIS-040A	In normal mode a component can be selected by clicking on it.
UIS-040B	A selected item is outlined and the properties can be changed in the properties window.
UIS-040C	A selected item can be moved by dragging.
UIS-040D	In normal mode a context-menu appears when pressing right-click on a component which gives the following options to the component: duplicate and delete.
UIS-050	The draw mode can be selected by pressing .
UIS-050A	The draw mode allows to draw a pipeline between an unused output of a component to an unused input of another component.
UIS-060	A pump can be added to the network by selecting and then the drawing.
UIS-070	A sink can be added to the network by selecting and then the drawing.
UIS-080	A merger can be added to the network by selecting and then the drawing.
UIS-090	A splitter can be added to the network by selecting and then the drawing.
UIS-100	After a component is added to the flow network the mode is set back to normal.
UIS-110	The sidebar has a properties box which allows to edit the properties of the a selected component.
UIS-120	The flow network is shown in the drawing.
UIS-130	Each component displays the current flow except for splitter and merger.
UIS-140	Components that have missing connectors on the input or output are displayed in orange.
UIS-150	Pipelines that exceed the maximum flow are shown in red.

3 | Nonfunctional Requirements

Code	Requirement
NON-010	The program has an user friendly design.
NON-020	The software is made testable
NON-030	The program has good performance whereby the application is responsive.
NON-050	The program is always available when the user needs it.
NON-060	The program is maintainable.
NON-070	The program is efficient so the task can be done as quick as possible.
NON-080	The program is stable, the program doesn't show any errors.
NON-090	The program has a good user experience which means it is a pleasant working experience for the user.

3.1 Usability factors

Code	Requirement
USF-010	The application is designed to be used with a mouse.
USF-020	Buttons have appropriate sizes to avoid mis-clicks.
USF-030	The colours used for representing different meanings should have clear definitions.

4 | Decisions and Rules

1. Algorithm chosen

   We choose to use the mathematical way to calculate the flows. The reason is that it is easier for us to do the calculation and programming, though in the reality, the way the system works may seem bizarre. For example when an output is not connected to a splitter the flow is stil mathematically divided.

2. Default values

   We set the default numbers of pipes and pumps to 0, and the default value of each splitter to 50/50. However, the value of each splitter is adjustable. With these default settings, our system has a clearer starting point since relative information is not mentioned in the requirement. Since an normal splitter can also be achieved with an adjustable splitter it is easier to remove the normal splitter to prevent confusion.