System Specification

Extended Petri net Simulator

Group A

October 25, 2013

Abstract

This document includes the requirements for the Software Engineering 2 project, which consists of an Extended Petri net software system.

Contents

1 Introduction

2 Overall description

2.1 General product description

2.2 Basic functionality

3 System Features

3.1 Petri net editor

Author: ***Albert***

The Petri net editor is a component that will extend the features provided by the *ePNK*, in order to fulfil the requirements of the project. The extended features include *Input Places* and the definition of links between geometry and animation components.

3.1.1 Functional Requirements

1. The Petri net editor **shall** allow the user to create, edit and delete Places.
2. The Petri net editor **shall** allow the user to create, edit and delete Tokens inside Places.
3. The Petri net editor **shall** allow the user to create, edit and delete Transitions.
4. The Petri net editor **shall** allow the user to create, edit and delete Arcs. An arc **shall** connect a Place to a Transition or vice versa.
5. The Petri net editor **shall** allow the user to define a Geometry label to a Place.
6. The Petri net editor **shall** allow the user to define an Appearance label to a Place.
7. The Petri net editor **shall** allow the user to define an Input Place label to a Place.
8. The Petri net editor **shall** allow the user to define an Animation label to a Place.
9. The Petri net editor **shall** allow the user to save and load a Petri net model.
10. The Petri net editor **shall** create a Petri net file in a format that can be read by the Simulator.
11. It **would be nice** that the Petri net editor allowed the user to undo and redo actions.
12. It **would be nice** that the Petri net editor allowed the user to copy and paste.

3.1.2 Use cases

The features are shown in Figure 1.

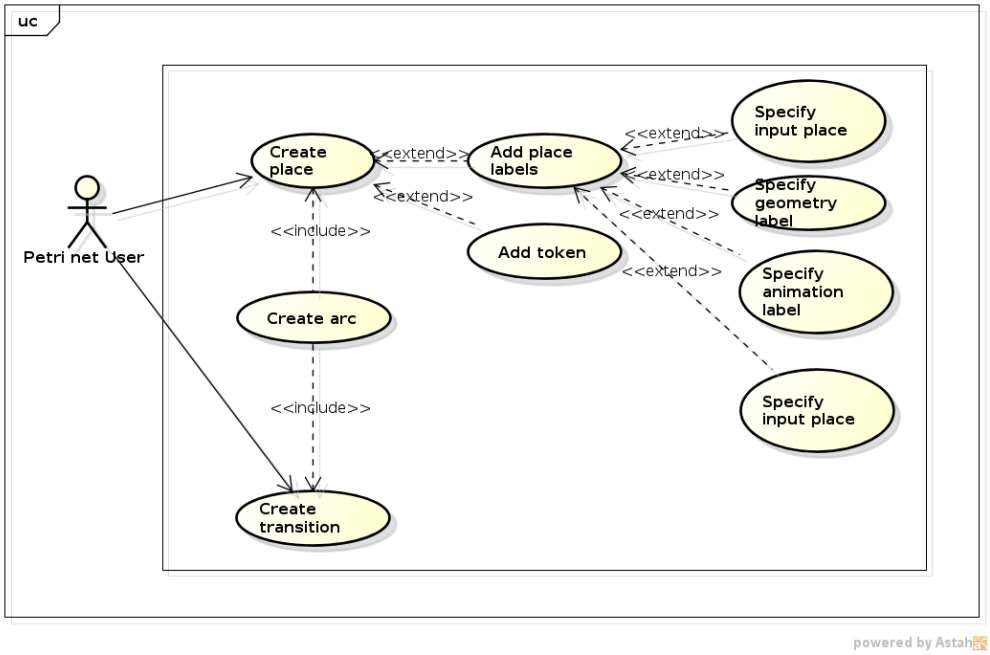

Figure 1: Use cases for the Petri net Editor

3.2 Geometry editor

3.3 Appearance Editor

3.4 Configuration Editor

Author: ***Albert***

The configuration editor is a component that will work as a connector between the Petri net editor (Section 3.1), the Geometry editor (Section 3.2) and the Appearance editor (Section 3.3).

3.4.1 Functional Requirements

1. The configuration editor **shall** allow the user to input a Petri net file containing its model.
2. The configuration editor **shall** allow the user to input a Geometry file containing its model.
3. The configuration editor **shall** allow the user to input an Appearance file containing its model.
4. The configuration editor **shall** allow the user to start a simulation with the referenced Petri net, Geometry and Appearance models.
5. It **would be nice** if the configuration editor allows the user to validate the data.
6. It **would be nice** if the configuration editor to save and load a specific configuration to a file.

3.5 Use cases

The features are shown in Figure ??.

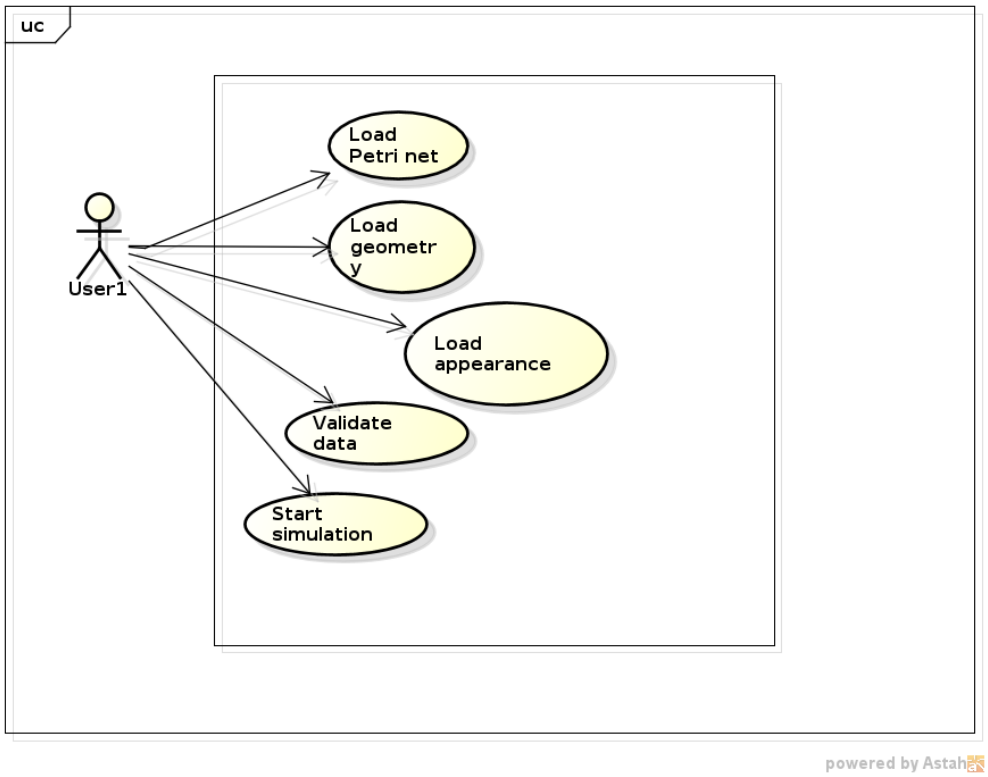

Figure 2: Use cases for the Configuration Editor

3.6 3D Simulator

4 Non functional requirements

4.1 Implementation constraints

4.2 Documentation

4.3 Quality Assurance

5 User Interface

5.1 Technology

5.2 GUI parts

5.3 Handbook

6 Architecture

7 Glossary

GUI Graphical User Interface

Petri net A mathematical and graphical model for the description of distributed systems. It is a directed bipartite graph, in which nodes represent transitions and places. The directed arcs describe which places are pre- and/or postconditions for which transitions. [source wiki]

Synchronization Transition finings are synchronized on the occurrences of external events, such as when animation is finished or user triggers the transition.

Token Petri Net element which moves along the Petri net places through transitions.

Use case A list of steps defining interactions between a role (e.g. "Technical user"), also known as an actor, and a system for achieving a goal. The actor can be a human or an external system

3D Three dimensional.

ePNK Eclipse Petri Net Kernel.