Software Engineering 2: Project Teamwork
Project plan
A visual simulator for high-quality planning domain models
Mälardalen University

Group 2

January 11, 2019

Contents

List of Tables

List of Figures

1 Introduction

The main goal of the project is to deliver an interactive graphical presentation tool that will help visualize and better understand the state flow of PDDL (Planning Domain Definition Language) domain models. The project will be delivered to the responsible person from the Ericsson company, Swarup Kumar. To do this, we will create a tool where the user can import a textual PDDL domain model and simulate it for debugging purposes. The GUI will show the veracity of the predicates and indicate which actions are possible. By selecting one of the possible actions, the state tree will update, showing us the changed state. From there, we can go back to the previous state (undoing the action) and choose a different action. The chosen software development process is the agile model since the project is characterized by an incremental approach, where we regularly discuss our prototype with the customer company, Ericsson.

2 Project Organization

2.1 Project group

Allan Kamuran: akn14006@student.mdh.se
Erik Eriksson: een14007@student.mdh.se
Erik Pettersson: epn14012@student.mdh.se
Ermal Bizhuta: eba17001@student.mdh.se
Johannes Chamuon: jcn16005@student.mdh.se
Laura Toric: ltc18001@student.mdh.se
Marta García Marín: mgn18015@student.mdh.se

2.2 Steering Group

Jan Carlson: jan.carlson@mdh.se
Robbert Jongeling: robbert.jongeling@mdh.se

2.3 Client

Company: Ericsson
Contact person:Swarup Kumar Mohalik. Location: New Delhi, India.

2.4 Roles and responsabilities

Allan - Configuration Manager (Trello)
The Configuration manager for Trello is responsible for managing the Trello board for the project. The Configuration manager also divides work tasks into smaller parts and assigns the team member to the tasks. Furthermore, deadlines and making sure the plan is followed are also responsibilities of the role.

Erik E - Configuration Manager (Development studio)
The role of the configuration manager for the development studio is responsible for the structure of the overall project in the Integrated development studio (IDE).
Erik P - Presentation Overseer
The presentation overseer is responsible for creating the PowerPoint presentations for each meeting with the steering group and the three major presentations for the project. Note that the rest of the group will also be involved in validating and editing the PowerPoint presentations.
Ermal - Group Manager
The group manager's responsibility is to manage the group and to lead the project as a whole. The group manager also validates tasks before the configuration manager assigns them to the group.
Johannes - Client Contact
The client contact's responsibility is to contact the client whenever it is needed. If the group want to ask the client something, then communication will go via the client contact.
Laura - Configuration Manager (GIT)
The role of the GIT configuration manager is to manage and organize the GIT repository. Additional responsibilities include making sure that all members of the group understand how to utilize GIT and help out with GIT related issues if needed.
Marta - Documentation and Report Overseer
The role of the documentation and report overseer is to make sure that documents are structured properly, are free from grammatical errors, and that information is correct and relevant.

2.5 Group organization and communication

We have planned meetings every Wednesday and Friday at 13:00. Every Friday we also communicate with our contact person via Skype to discuss the details about the project as well as solve any uncertainties about the project.
On Wednesdays, we plan on discussing problems that we have encountered and as well check up on work that needs to be done that week. If we stumble upon emergent problems that we need to consult with the client, we can contact him on this day as well.
On Fridays, we will discuss the completed work of the week, validate the project and prepare for the next week's sprint (Sprint review + Sprint retrospective + Sprint planning).
The complete schedule of our working routine can be found on Trello, as well as the work in progress and the upcoming tasks to fulfill. Trello helps the planned work to flow more efficiently and allows us to get a better visualization of all the members' responsibilities for the project.
To divide the responsibilities into the project's tasks, we've assigned them to each team member by linking the task to the member's account. This allows us to see what tasks each one performs, and to monitor and confirm the attendance

of team members at our weekly meetings.
When a task is to be done, we add it to the sprint backlog column, after assigning the members working on it. When the task starts, it should be moved to the progress column, so the rest of the team will know what activities are ongoing and who is doing them. After the activity is completed, we will move it to the finished column, adding the files as attachments and then uploading them to the git.
As a communication tool, we have made a Discord channel where we talk about details regarding meetings and work to be done. Discord allows to have multiple servers, so we have a channel for report talk, one for coding talk and a general one.
Once the implementation begins we will use GitHub to work on the project as a group. All the project code and documentation files will be available there.

2.6 Outlook of working hours

Every week we will have 3 team meetings of 2 hours each, which means 6 hours of teamwork per member. Besides this, we will have an hour of meeting with Jan and another one with Swarup to discuss the development of the project and possible doubts that arise during the process. If necessary, we will have an additional meeting with Swarup by contacting him via Skype.
Finally, we have established a personal working time of between 8 and 12 hours a week. So far we can assume and predict a total sum of weekly work that will vary between 16 and 22 hours of work on the assignment per member every week. This will depend on the work to be done each week according to the project phase (design, testing etc.) and the responsibility of each one in the activity to be performed.

2.7 Activity Plan

We have created a Gantt chart with the weeks needed for every activity. The Gantt chart below shows the duration, start and end of every task. Some tasks overlap with each other because they run simultaneously. Most of the overlapping happens between design and implementation since we will implement an initial version of the tool and then improve the design of the tool according on the changes to be made discovered during the implementation phase. The activity plan is shown in the Figure 1

2.8 Quality assurance

To assure high prod[a]uct quality we plan on having weekly meetings with our contact person in Ericsson, Swarup, where we will discuss the project in detail to make sure we are building the right product and fulfill the requirements as well. Besides that, every team member will have 2/3 other team members responsible for reviewing their code to make sure no bugs or improper coding practices are delivered to the client.

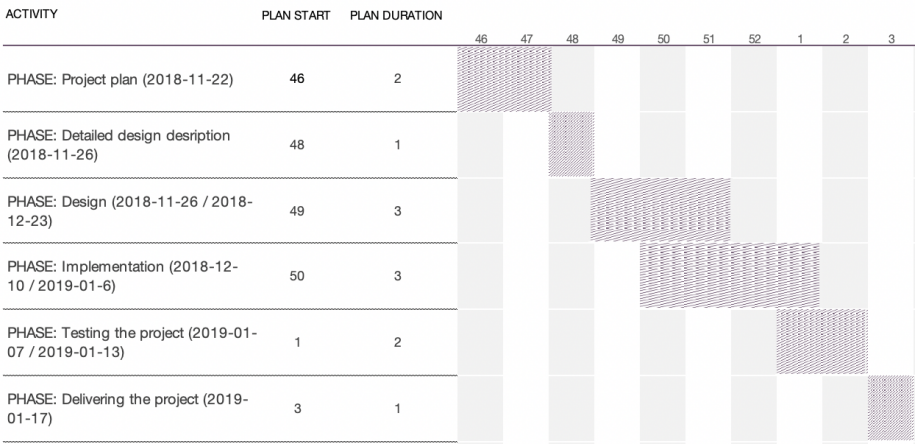

Figure 1: Activity Plan

2.9 Initial Backlog

For this project, we have a few different kinds of backlogs. The different backlogs are Product backlog, Meeting backlog and Sprint Backlog. We have a few more columns like In progress, Completed and an Archive column for each week. But for the initial backlog, we have a few big tasks that will be worked on and split up as we go with the project.

Project Plan
Send in a sort of Project Plan for the project. We will split up the project plan into more smaller parts to everybody in the group will have tasks to do.

Class presentation 1
For this task will we have the first class presentation and will have a 20-minute presentation about the Project plan.

Design and Implementation
For this part will the design and implementation part of the project start. We will split up this big task into much smaller parts and will have a significant consideration about how in the group knows what languages to everybody is comfortable with their specific task.

Class presentation 2
For this task will we have a second class presentation and will have a 20-minute presentation about the Design and implementation phase of the project.

Test and delivery
In this task will we test the product and when it is finalized will we then and only then send it to the client.

Class presentation 3
For this task will we have a third class presentation and will have a 20-minute presentation about the test and delivery phase of the project.

3 Description of the system to be developed

3.1 High-level description of Artificial Intelligence Planning and PDDL

Today, artificial intelligence (AI) is an emergent field in Computer Science thanks to technological advances, making it possible for computers to run complex algorithms that require a lot of computing power. The need for AI planning comes from an increasing trend of automation in many fields of business and research. Robots and autonomous vehicles are examples of AI agents that can use AI to solve complex problems. AI planning makes it easier for people to understand the agent choices in taking particular actions. PDDL is a simple language that helps efficiently utilize AI planning.
PDDL consists of two parts, the domain description and the problem description. The domain description specifies how the world functions. Logical facts called "predicates" specify for example where an object is in the world. The domain description also specifies how we can interact with the objects using actions. The actions contain preconditions that need to be fulfilled in order to perform the action and effects that specifies how the world changes when performing the action. The problem description specifies the objects that are present in the world and what the goal state is for the problem. When the goal state is reached the problem is solved.

3.2 Project Description

Since planning is a significant research and application area in AI, our product should be able to help simplify the development of high-quality planning models. The project aims to build a tool where the user can either import or edit a textual PDDL domain model, and simulate it for debugging purposes. Debugging PDDL domain models is useful for detecting and avoiding deadlocks when it comes to AI planning problems. The tool that will be developed will provide a visual and textual simulation GUI that simplifies the debugging process by presenting the current state and the actions that are possible. The visual simulation of the states provides the user with an overview of how any potential deadlocks stages have occurred. When selecting one of the possible actions, the state should be updated. The tool should also provide the functionality of returning to a previous state, and resume the debugging from there.

3.3 Requirements

The requirements table is shown in the Table 1

3.4 Use cases

3.4.1 Use case diagram

The use case diagram is include in the Figure 2

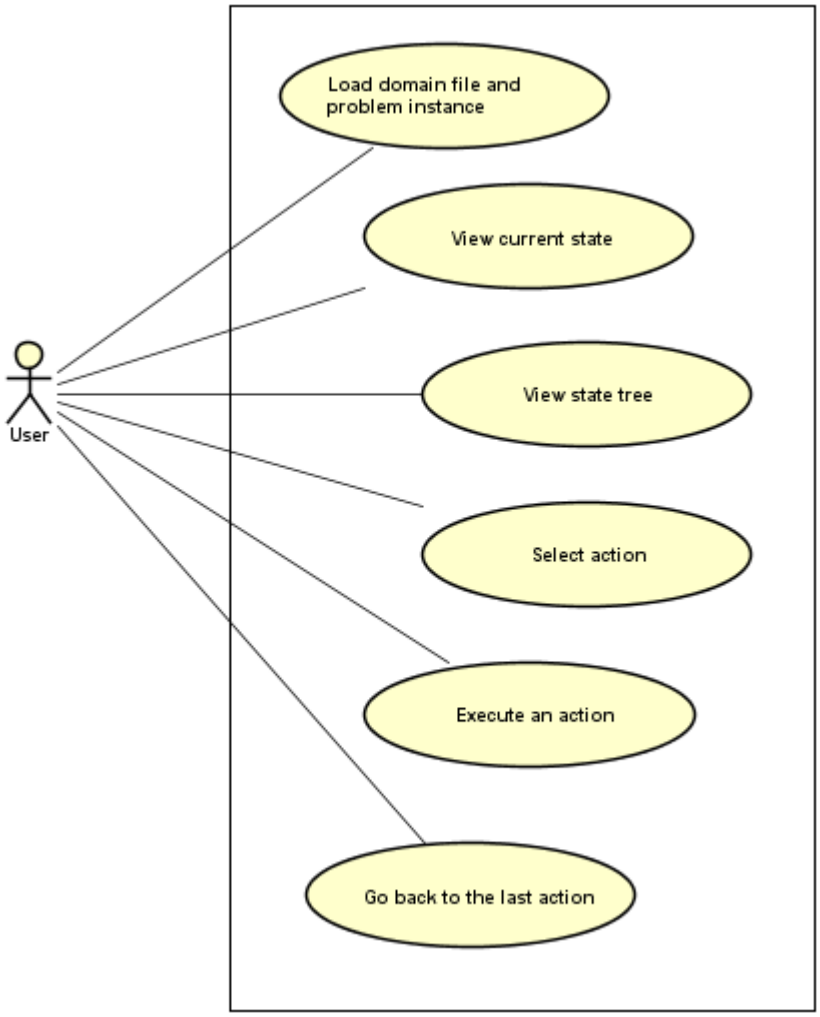

Figure 2: Use Case diagram

3.4.2 Use Case 1: Load domain file and problem instance

Name: Import
Initiator: User
Goal: Import the domain file and problem instance to edit or simulate/debug.
Main Scenario:

1. User import the domain file
2. problem instance

Extensions:
Imported files are in the wrong format:

a) The system throws an ERROR
b) Resume at 1

3.4.3 Use Case 2: View the current state

Name: View the current state
Initiator: User
Goal: The user selects the current state, being able to see how all the predicates are in that moment
Main Scenario:

1. The user selects the current state
2. The system shows the most important predicates for that state
3. By clicking a button, the user is able to see all the others states and predicates in the problem

3.4.4 Use Case 3: View state tree

Name: View state tree
Initiator: User
Goal: The user should be able to see the hole state tree of the states he has been throw.
Main Scenario:

1. The user selects the option of viewing the state tree
2. The state tree appears with all the states the user has been throw and all the actions he has been able to choose. If the state tree is bigger than the screen, there will be a scroll bar in the right side of the screen.
3. The user is able to choose a state in the state tree so he can view it.

3.4.5 Use Case 4: Select an action

Name: Select an action
Initiator: User
Goal: The user chooses an action from the state tree.
Main Scenario:

1. The state tree appears with the position of the user in it
2. The user selects an action between the options he can take from his position
3. The system visualizes the selected state

3.4.6 Use Case 5: Execute an action

Name: Execute an action
Initiator: User
Goal: The user executes an action
Main Scenario:

1. The user selects an action from the state tree
2. The user executes the action previously selected
3. All the attributes from the first actions that are different than the ones in the action selected change, so the current state is updated

3.4.7 Use Case 6: Go back to the last action

Name: Go back to the last action
Initiator: User
Goal: The user goes back to the last current state, undoing the action made.
Main Scenario:

1. The user previously selects an action
2. The user selects the button to go back, so he can return to the last current state, and it is not recorded in the state history

Priority	Description
High	**Importing PDDL** The user shall be able to import a textual PDDL domain- and problem description into the application.
High	**Parsing domain- and problem definition** The system shall synthesize (and simulate) the two PDDL files and parse them for debugging purposes.
High	**Visualizing the domain model** The user should be able to visualize the extracted data in textual and graphical form.
Medium	**Simulating the domain model manually** The user can choose an action and the current state and the state tree shall be updated.
Medium	**Returning to the previous state** The user can choose to go back to any previously visited state and continue from there.
Medium	**Simulating the domain model automatically** The system can select arbitrary actions and simulate the domain model.
Low	**Visualizing the states' tree** The user should be able to visualization of the whole tree of reached states.

Table 1: Requirements