

CSCI 477 Class Project – HazMat

Spring 2003



1. Objective

The Hazardous Material Rover [HazMat] project provides you the opportunity to practice the software engineering principles you learned in the CS 477 class. HazMat is a project whose goal is to engineer a robot remote control simulation system. The project simulates a class of emerging real-world situations in which humans prefer to use intelligent, mobile, possibly remote-controlled robots in search-and-rescue missions, deep sea exploration, hazardous waste disposal, and so forth. As such, the HazMat project involves different key stakeholders.

Table 1 HazMat Key Stakeholders

<i>Client</i>	USC Center for Software Engineering
<i>Developer</i>	CS 477 team
<i>Operator</i>	An agent of our client responsible for operating the HazMat rover and maintaining the external “environment.”
<i>Configuration Manager</i>	An agent of our client responsible for preparing the HazMat rover for each mission and deploying it into place.

Your mission is to complete the project and satisfy the system requirements. You are required to study the project context, understand the win condition of each key stakeholder and the project requirements, design the HazMat system, implement a prototype of it, and test it on a simulator. You will follow the MBASE approach to navigate through your project process. You will be provided the necessary project documentation (in the form of the MBASE OCD and SSRD documents). However, it is important to keep in mind that these documents may not be “perfect”; it is your task as a software development team to interact with your customers and make sure that any inconsistency, incompleteness, and ambiguity in the documents is resolved.

2. HazMat in a Nutshell

In environments that have been rendered unfit for human presence by a catastrophe of some kind, there exists a compelling need for a remote-controlled rover to document objects of interest (including hazards and obstacles). This rover will be equipped with an aimable camera and will act as its operator’s “eyes”, allowing the operator to record the location, configuration, and image of each object of interest.

In this application context, the HazMat Rover is being developed for the Center for Software Engineering (CSE) as a component of a larger project sponsored and led by NASA. HazMat consists of two components:

1. the robot agent, which is simulated by the “player” and
2. the environment, which is simulated by the “Stage”, as illustrated in Figure 1.

In the *Stage*, we are able to define the size and boundary of the environment, program the obstacles, and set the objects of interest. The *players* are able to accept controlling commands from client software such as going ahead, changing direction, switching speed, and so on. Each *player* is equipped with a camera. Through the camera, the operator can see the images captured by the camera. Both the *Stage* and the *Player* can be implemented using the Pioneer 2 APIs, which will be discussed further later in the semester. For more information, you may also refer to <http://playerstage.sourceforge.net/>.

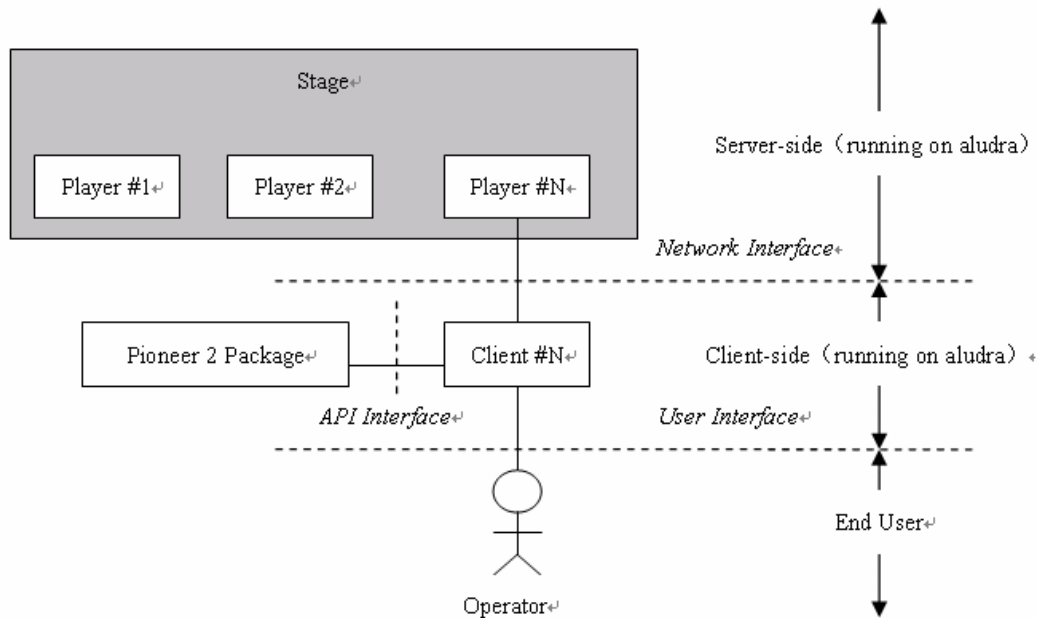


Figure 1 The framework of HazMat

3. Project Requirements

- ♦ **Personnel** – The project will be undertaken by 5-person teams. Once you are assigned to a team, you are expected to get familiar with your teammates, divide the job, and coordinate your work.
- ♦ **Schedule** – The project schedule should be within this semester, as stated in the course Syllabus.
- ♦ **Budget** – There will be no monetary budget available for the class projects. Alternatively, you may think of yourself as having an unlimited budget involving the imaginary money of your choice. ☺
- ♦ **Desired system features** – The desired **core** system features may include:
 - Exploring an area based on an existing map – The robots should be able to move in different environments based on the provided (fixed and complete) map files.
 - Reaching pre-defined spots without an existing map – The operator may want the robot to visit a pre-defined spot and gather information, e.g., by taking a photo at that spot. The robot should be able to explore the path autonomously.

Other system features may include

- Dealing with an imperfect map – If the provided map is incomplete, the robot should be able to react to the discrepancies (e.g., an unexpected obstacle) and autonomously discover alternative paths.

♦ **Project Deliverables**

Your project package should include the following artifacts:

- MBASE System and Software Architecture Description (SSAD).
- The source code of your prototype in a language of your choice. The player/stage supports C/C++, Java, and Python.
- Related project documentations.

Note: This document is a only brief project description, providing some examples to give you a flavor of the project. These are *not* the project requirements. For more detail about the requirements, you will need to study the provided project reference documents, and communicate with the project customers.