CM30174/CM50206: Intelligent Agents:
Assessed CourseWork 2: Planetary Rovers

Julian Padget
jap@cs.bath.ac.uk

November 4, 2013

1 Introduction

The goal of this course work is to design and implement agents that explore a simulated environment and to create an ontology that allows them to communicate with each other to achieve their goals. This work needs to be carried out individually.

2 Objectives

At the end of this course work you will be able to

- develop agents using the Agentscape platform
- develop an ontology
- build on existing code
- explain and evaluate design decisions and implementation in a concise manner

3 Project

The following subsections provide the project description, functional and non-functional requirements, the guidelines for installing the necessary code to start your project and the requirements for the report to be submitted for assessment.

3.1 Background Information

The situation is a simulated planetary exploration scenario where you must develop "rover" agents that explore an environment, locate resources, and then return them to a base.

Your agents will attempt to complete a number of different scenarios. You may create a single agent that is used in all scenarios or different agents for different scenarios. For each run of a scenario:

- Rover agents start at a base at a randomly selected location.
- Each agent is given a limited amount of energy.
- Resources are scattered throughout the environment.

The rover agents must explore the environment and return as many resources to the base as possible. Agents that run out of energy can no longer function. When all agents run out of energy, or all resources have been collected, the scenario is complete.

The Basic System You are provided with a basic system to begin the coursework. This system consists of a service for Agentscape that controls the environment and simulation, a simple rover agent that interacts with the service, and a monitor agent that controls the simulations and displays information about the environment.

The rover service is an Agentscape service that controls the environment and keeps track of the agents in the simulation. The basic agent communicates with this service and provides methods through which your agent interacts with the environment, such as "move", "scan", "collect" and "deposit". You extend the basic rover and implement more complex behaviours using these primitive methods.

3.2 Functional Requirements

Your task is to:

1. Extend the basic rover agent to create an agent or agents that can successfully gather resources in the different scenarios.
2. Create an ontology that allows your agents to communicate with each other during the multi-agent scenarios. This should be done using Protégé and the Ontology BeanGenerator plugin.

3.3 Non-functional Requirements

1. The program must be entirely written in Java.
2. The program should be written in a object-oriented fashion. Methods consisting of more 30 lines of code will be looked at with suspicion.
3. The program should work correctly on the BUCS Linux (lcpu) system.
4. Only the standard Agentscape platform may be used in conjunction with the basic rover agents and rover service provided.

3.4 Software Supplied

The following components are made available for download:

1. Agentscape 2 Milestone 2 (download from `agentscape.org`)
2. The Protégé ontology editor (version 3.4.4, download from `protege.stanford.edu`)
3. The Ontology BeanGenerator plugin for Protégé (version 3.4, download from `http://protegewiki.stanford.edu/wiki/OntologyBeanGenerator_4.0`)
4. Basic rover agents and the rover service (available on Moodle)

3.5 Scenarios

Your agents will be run in a series of tests with different conditions to determine that they act appropriately in each situation. These scenarios are:

1. A single agent with one resource location. High energy limit.
2. A single agent with several resource locations. High energy limit.
3. Three agents with many available resources. Medium energy limit.
4. Three agents with sparse resources. Medium energy limit.
5. Ten agents with sparse resources. Low energy limit.
6. A competitive scenario where five agents from each student share the same world. Medium energy limit.

3.6 The viva-voce

Your mark will be determined in part by the performance of the code you submit and in part by your performance during a viva-voce of 15 mins. The latter will address some of the following issues:

1. A detailed discussion of the different agents you implemented

2. A justification for the decisions made
3. A critical analysis of your implementation
4. Directions for further improvements
5. Evidence and discussion of test case results

4 Assessment

4.1 Conditions

The coursework will be carried out individually. Attention is drawn to the University and Departmental rules on plagiarism (page 49 of the Programmes Handbook). The coursework can be worked on during the tutorial sessions. The coursework also involves work during one's own study time.

4.2 Marking

This coursework counts for 25% (20% for MSc students) of your final mark for this unit. Marks are awarded on the basis of the viva-voce and the running of the submitted code against the test cases identified earlier. The breakdown of marks follows:

- **50 percent** – Viva-voce.
- **10 percent** – Single-agent scenarios.
- **15 percent** – Three-agent scenarios.
- **20 percent** – Ten-agent scenarios.
- **5 percent** – Competitive scenario.

The marks for a scenario will be avarage score of running the scenario 5 times. The following is a guideline so that you are aware of what we expect during the viva-voce:

- 1st. The viva-voce reaches the standard for a 2:1 student. In addition, the students demonstrates a deep understanding of the problem and pulls in ideas from background reading. Information is presented in a clear and concise manner and shows a clear appreciation of other areas of knowledge and sources of expertise that might be brought to bear on the coursework.
- 2:1. The coursework idea is clearly well understood. Evidence of the identification of background reading sources is shown. Rationale is presented in a concise manner without much probing.
- 2:2. The student demonstrates and explains how he/she expanded/modified the original coursework idea to suit their own skills and understanding. Some deeper understanding of problem domain is shown. The description of the agent behaviours and justification of decisions made demonstrates effort to make the different kinds of agent meet the requirements in the time-frame of the coursework.
- 3rd. The student is hardly able to explain the supplied basic rover agent and the description of agent behaviours and ontology demonstrates a lack of understanding of the code that was provided. Very little understanding is demonstrated on how to move beyond the code provided in order to achieve the various scenarios.

4.3 Deadlines

The coursework is published on **Tuesday 5 November, 2013 (week 6)**. The deadline for submission is **Tuesday 10 December, 2013 midday (week 11)**. Vivas will take place during the week beginning **Week 12**.

4.4 Coursework Submission

You should submit your coursework using the unit's Moodle page (`http://moodle.bath.ac.uk/moodle5/course/view.php?id=400`). You should submit your report in pdf format along with a zip file containing:

- A directory containing the source code for your agents.

- Jar file(s) containing the compiled code for your agent(s).

4.5 Estimated Workload

The coursework is expected to take about 20 hours. However, programming is a skill. Different students will spend very different amounts of time when it comes to programming.

4.6 Feedback

Feedback will be given in the form of individual written comments on Moodle within three weeks of the viva voce. Clarification of this feedback can always be obtained from the unit lecturer(s).