Teaching, Learning and Assessment of Agents and Robotics in a Computer Science Curriculum

Ioanna Stamatopoulou, Konstantinos Dimopoulos, and Petros Kefalas

The University of Sheffield International Faculty, CITY College
3 L. Sofou, 54624, Thessaloniki, Greece
{istamatopoulou,k.dimopoulos,kefalas}@city.academic.gr

Abstract. This paper presents our experience in integrating agents and robotics in our Computer Science Curriculum. We present a series of modules throughout our curriculum that progressively address these themes and other AI related topics, which ends with a specialised final year module central to teaching and learning multi-agent systems and principles of robotics. As part of this module a Robotics Challenge is organised, allowing students to integrate the knowledge they obtained in previously attended modules, and to practically apply knowledge and skills in order to solve a real problem.

Keywords: Agents, Multi-Agent Systems, Robotics, Computer Science Education

1 Introduction

Agents, Multi-Agent Systems (MAS) and Robotics have gained increasing attention in research, mostly related on how they can be designed implemented in order to exhibit intelligent behaviour. We believe, however, that topics could be more extensively be incorporated in Computer Science Curricula in order to better prepare students towards the design and development of distributed, smart, complex systems, when they face these challenges in their professional careers.

The Joint ACM/IEEE Task Force includes *"Agents"* in its 2013 version of Computing Curricula[1] within different Knowledge Areas, such as HCI, Intelligence Systems, and Social Issues and Professional Practice. *"Robotics"* is included in both Intelligent Systems and Platform-Based Development. In the 2012 ACM Computing Classification System,[2] the topics are listed in Computing methodologies, Artificial intelligence, Distributed Artificial Intelligence as well as in Computer systems organization and Embedded and cyber-physical systems, respectively. As such, universities have MAS in their programmes either as a separate module or incorporated within more generic modules, such as the ones listed as Course Examplars.[1]

[1] http://www.acm.org/education/CS2013-final-report.pdf
[2] https://www.acm.org/publications/class-2012

Our aim in this paper is to report our experience from teaching a number of modules with agents, MAS and Robotics as a main theme. We provide an overview of how these topics are covered throughout our curriculum and focus on the design of a final year module, the concepts we introduce on software and robotic agents, and how we assess students. The feedback we have so far is extremely positive and we are prompted to encourage colleagues to borrow and expand our ideas for their own teaching, learning and assessment methods.

The rest of the paper is structured as follows: Section 2 briefly presents a series of modules throughout our curriculum that progressively address agents, MAS and Robotics, while Section 3 focuses on the module Agents and Robotics that is offered in the last semester of our Bachelor's programme. Section 4 discusses in detail a Robotics Challenge that is organised as part of this module and Section 5 evaluates our approach by presenting the results of a questionnaire completed by students. Finally, Section 6 concludes the paper.

2 Agents in Computer Science Curriculum

In our curriculum we cover agents, MAS and Robotics from various perspectives throughout all three levels of the Bachelor's degree and at Master's level, aiming towards a gradual enhancement of skills. Table 1 provides a summary of all related modules' aims, along with their European Qualifications Framework (EQF) levels.[3]

In the first year of studies in the Programming Principles and Algorithms module a reactive agent platform is used to introduce basic programming skills, through the use of a tool called Mentor[4] that facilitates the visual outcome of a programmed reactive agent. The concept is similar to that of RoboMind Academy,[5] with the difference that the programming language of Mentor is Java. The environment provides a two-dimensional space in which an agent/virtual robot nay be programmed to perform simple problem solving tasks by perceiving and affecting the environment (Fig.1).

In the second year of studies the Artificial Intelligence Techniques module teaches students structured knowledge representation techniques, and search and constraint satisfaction algorithms for problem solving. Although these are fundamental principles of the broader AI field, we take the opportunity to discuss how the world and a problem domain could be represented to form an agent's knowledge and beliefs, and what kind of techniques agents may employ to reason about their state and the world.

In the third year of studies the module Intelligent Systems comes as a sequel of AI Techniques and expands on a variety of AI areas, aiming at breadth instead of depth. Students are exposed to how neural networks and stochastic systems, fuzzy reasoning, planning, and learning can facilitate the reasoning of intelligent

[3] http://www.accreditedqualifications.org.uk/european-qualifications-framework-eqf.html
[4] http://robotseducate.us
[5] https://www.robomindacademy.com

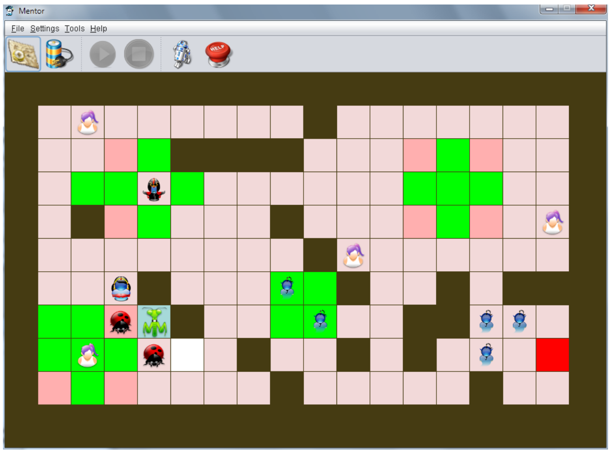

Fig. 1. The tool used in our Programming Principles and Algorithms module that allows students to program a virtual robot.

agents. Also in their third year students attend the Agents and Robotics module which is extensively discussed in Section 3 of this paper.

Finally, postgraduate students, irrespectively of their background, attend a module entitled Knowledge Technologies for Innovation. The module discusses how knowledge technologies may be exploited so as to increase the performance of classic enterprise systems and facilitate quality decision-making aiming towards product/service innovation. The module is an opportunity to demonstrate a good number of applications that stand at the frontiers of innovative smart business management products. The interested reader may find more information about this module in [7].

3 Agents and Robotics

The Agents and Robotics module is taught in the last semester of the undergraduate studies and offers students a unique opportunity to wrap-up the knowledge and skills they have acquired in all previously agent-related attended modules, and apply them in one coherent application, i.e. a robot.

This module aims to:

- introduce students to various types of agents, and their architecture, strengths and limitations;
- introduce multi-agent systems, agent communication and interaction;
- discuss possible application areas of the intelligent agent technology through examples and case studies;
- discuss the advantages of the agent-based approach to engineering complex software systems;
- introduce students to mobile robots, the related issues involved, and their applications;

Table 1. Agent-related modules' aims and EQF levels

Module	EQF Level	Overall Aims
Programming Principles and Algorithms	5	To introduce problem analysis, algorithmic thinking, and design practices, such as incremental code writing.
Artificial Intelligence Techniques	6	To introduce main principles of AI: knowledge representation techniques, reasoning and search algorithms, as well as principles of natural language understanding.
Intelligent Systems	6	To expand to more specific areas of AI, such as neural networks, fuzzy systems, planning, and machine learning, as well as their main applications.
Agents and Robotics	6	To introduce fundamentals of intelligent agents, multi-agent system design, principles of robotics, through hands-on implementation of robotic agents
Knowledge Technologies for Innovation	7	To provide an overview of knowledge technologies, accompanied by a series of case studies, demonstrating their applicability on smart systems and their potential for business innovation.

- investigate robotic technologies relevant to sensing, perception, action and re-action;
- discuss the evolution of robotics in the immediate future, and determine innovative applications;
- underline the similarities and differences between software agents and mobile robots.

We use Bloom's taxonomy [2] to describe the cognitive level of the learning objectives, and we expect that by the end of the module students are able to:

LO1 *explain* the basic notions of agent systems and the difference between agents and other programs;
LO2 *describe* the fundamental agent architectures and sensibly *design* reactive and BDI agents;
LO3 *discuss* the issues involved in designing multi-agent systems, particularly with respect to communication and interaction, and *apply* techniques for addressing them;
LO4 *demonstrate* an overall understanding of biology inspired agents;
LO5 *argue* that the agent paradigm is an alternative point of view to software engineering and *realise* the related agent-based software engineering methodologies;
LO6 appropriately *taxonomise* robots;
LO7 *explain* the problems involved in designing new robots regarding sensing and perceiving the environment, controlling the movement, and decision making;
LO8 *design* and *construct* simple robotic automata capable of performing simple behaviours.

Contents of the agent part of the module include: definition of the notion of agency, agents types and architectures (primarily reactive and BDI), multi-agent systems, agent communication and interaction, and biology inspired agents.

In the robotics part students are taught basic concepts, types and classification of robots, sensor types, robot movement and actuation, Kinematics of mobile robots, controlling motors and servos.

We use a variety of teaching methods ranging from lectures to workshops and training laboratories. For hands-on practice we use Netlogo and Lego Mindstorms as the tools with which students will practice their knowledge and skills. Netlogo [9] is a cross-platform multi-agent programmable modelling environment extremely suitable for MAS simulation. In NetLogo, the environment consists of a grid of patches and is inhabited by turtles: entities that operate in it, interact with it and among them. The effectiveness of using Netlogo in teaching agents is discussed extensively elsewhere [6][8] in which the interested reader may also find the reasons why other fully fledged tools for agent development are not preferred.

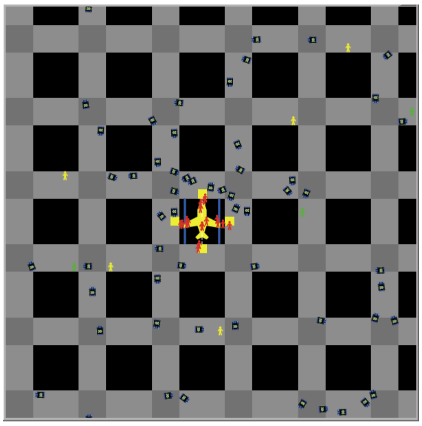

Fig. 2: Taxis negotiate and coordinate in order to carry passengers from any part of a city to the airport.

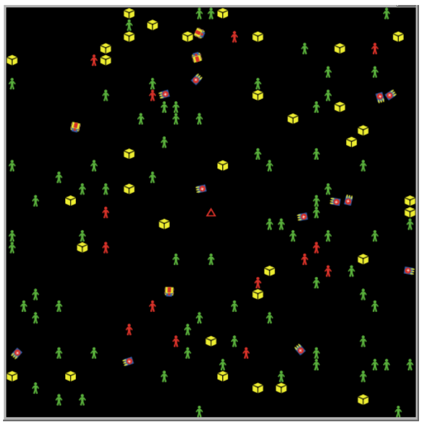

Fig. 3: Rescue units provide first aid to injured people they find in a disaster area. Ambulances collaborate in order to transfer the rescued people to the hospital.

Lego Mindstorms[6] is a versatile modular robotics platform, developed by Lego, aiming at a commercial and educational audience. We decided to use Lego Mindstorms for various reasons:

- **Modularity**: The Lego platform is inherently modular, allowing for a wide range of robotic constructs. Furthermore, the platform comes with a number of easy to use sensors that allow for a wide range of intelligent behaviours to

[6] https://www.lego.com/en-us/mindstorms

be developed, while third party sensors can also be used to further expand the versatility of the platform.

- **Ease of constructing custom robots**: Most students are familiar with Lego-type toys, and this helps them kick-start building a robot. In addition, many instructional sources are available online, to guide students towards building a variety of robotic chassis, or even inventing their own.
- **Ease of programming**: There is a variety of programming languages available for programming the platform, many of which are direct imports of known languages. LeJOS (Lego Java Operating System) is a Java import that is flexible, compatible with existing Java libraries, and comes with a number of libraries of its own that support agent-based concepts. The fact that our students our taught programming in the Java language makes LeJOS an appropriate choice.
- **Low cost**: The Lego Mindstorms platform costs a little over 300 euros per set, and includes all the basics needed. A single set may be used by a team of four to five students, making the total cost for a cohort of 20 students about 1,200 euro. There are not many educational robots that cost less that 1,200 per piece having the same flexibility.

The above reasons have made the Lego Mindstorms platform very popular in secondary and higher education [1][3][4]. The platform itself has been introduced in the 90s and has since been updated several times . So far, we have been using the NXT 2.0, as we have the sets available for the past 6 years, but we have lately acquired the new version of Lego EV3.

Both main themes of the module, agents and robotics, are assessed through two independent coursework assignments. The first involves the development of a MAS simulation in NetLogo (accomplishing learning outcomes LO1-LO5). This includes the design and implementation of independent agents that collaborate or compete to accomplish a certain task, such as carrying passengers to the airport (Fig. 2), rescue injured people in a disaster situation (Fig. 3), emergency evacuation of a building (Fig. 4), etc. Students are given libraries for BDI archirecture, FIPA exchange of messages and Contract Net protocol implemented in NetLogo [5]. The second coursework assignment involves the completion of a challenge on robot design and task fulfillment (accomplishing learning outcomes LO6-LO8), which is the focus of the following section.

4 Case Study: The Robotics Challenge

The Robotics Challenge event takes the form of a celebration in the Department of Computer Science and the Faculty as a whole. Many students and academic staff from other Departments are watching the setup, preparation and experimentation until the final demonstration. The challenge takes place over two full days with the final challenge taking place at the end of the second day.

Students are divided to teams of 4-5 and have already been introduced to Lego NXT 2.0 and its programming through a series of lab sessions. The number

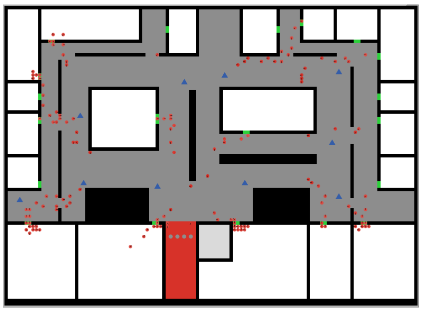

Fig. 4: People evacuate a building upon hearing a fire alarm, by following an exit plan located in each room.

Fig. 5: The terrain setup and the robotic agents operating in it.

of teams depends on the available resources and the number of students in the cohort (usually in the range of 20-25).

As an example consider the following challenge. The terrain the robots operate in is an enclosed area of 2×2 meters, surrounded by a short wall (Fig. 5). Two patches (A4 papers of different colours) are placed at specific parts of the terrain, representing the robots' nests. At random places in the terrain objects are placed representing food that can be picked up by the robots. These objects are cylinders of different colours with a diameter of 4.5cm and a height of 10.1cm.

The aim of the challenge is to create robots that will explore the terrain foraging for food. To guide students towards completing the challenge, it is broken into smaller ones so that the overall problem can be solved incrementally:

1. Move randomly inside the terrain avoiding obstacles (initially other robots, walls and food cylinders);
2. Explore the terrain looking for the nests, while avoiding obstacles (other robots, walls and food cylinders).
3. Differentiate between food cylinders and other obstacles (other robots and walls), and identify the cylinder's colour.
4. Explore the terrain looking for food cylinders and pick up them up (one at a time), avoiding other robots and walls.
5. Explore the terrain looking food cylinders and take them back to the nests, keeping the location of the nests and of the food cylinders it cannot pickup in memory.
6. Communicate the location of discovered nests and food cylinders that cannot be picked up to other robots.
7. Integrate all the above in creating a robot that explores the terrain avoiding other robots and walls, looking for nests and cylinders, reporting the location of those that cannot be picked up to other robots, and transporting food cylinders to the closest nest. Food cylinders are placed dynamically in random positions inside the terrain and are removed manually when they are placed at a nest.

Fig. 6: A final robot that takes place in the competition.

Fig. 7: Students working on the robotic challenge.

To complete the challenge students design and develop their own team robot (Fig. 6) and experiment with each of the aforementioned sub-challenges for two days (Fig. 7). Sub-challenges are first discussed with all the teams in a brainstorming session with the instructor, whose role is to coordinate the discussions, advise, assist with clarifications or feedback, and act as an expert reviewer.

Each sub-challenge is implemented as a new behaviour of the robot or as an enhancement on an existing behaviour. For example, for sub-challenge 2, which requires the identification of a nest, students need to consider the range of available sensors (camera, colour sensor, light sensor), design and implement the needed behaviour, and incorporate it with the existing obstacle avoidance behaviour of sub-challenge 1, by realising a subsumption agent architecture.

The final robots are ready for demonstration at the end of the second day. After the completion of the challenge, students submit a group report on how they developed their robots, and an individual report about their experience and their contribution to the group.

Throughout the years, we have come up with a number of different challenges. Indicatively, another challenge requires two kinds of robots: rescue-bots and carry-bots. Randomly placed at the grid there are civilians in danger. A rescue-bot locates civilian victims in need and provides them with first aid until the carry-bot arrives to transport the civilians back to the hospital (scenario very similar to the MAS simulation in NetLogo that we ask them to develop in their first assignment).

The most successful robot designs have been demonstrated by the students to the open public in two follow-up occasions: the 7th International Mathematics Week, organised by the Greek Mathematical Society,[7] and the 1st Thessaloniki Science Festival organised by the British Council.[8] Students had the opportunity to present their work to children in a simple manner so as to promote their interest in robotics and STEM subjects.

[7] http://www.emethes.gr [in Greek]

[8] https://www.britishcouncil.gr/en/events/thessaloniki-science-festival

5 Evaluation

By participating in the challenge, students are able to understand the problems involved in designing new robots regarding sensing the environment, controlling the movement, and decision making. They are also able to design and develop simple robotic automata capable of performing tasks of varying difficulty: from executing simple pre-programmed tasks to learning simple behaviours. The overall experience gives them the opportunity to consider agent architectures in designing a robot, and exercise the skills and knowledge acquired in other modules of the curriculum.

Students who have participated were asked to evaluate various aspects of the challenge as well as self-reflect on what they gained through the process. The questionnaire distributed was electronic and data collected were anonymous. Answers were in a 1-5 Likert scale or in a "Strongly Agree" to "Strongly Disagree" scale, whereas there was an opportunity for free text comments.

We had 100% Agreement in the statements:

- The Robotics Challenge was a positive experience;
- The timing of the Robotics Challenge was good, taking into account my other study obligations in the Department;
- The Lego platform used for the Robotics Challenge was appropriate;
- LeJos used to program the robot for the Robotics Challenge was appropriate.

Students also found that the Robotics Challenge helped them understand the theory (score 4.6/5) and that their programming skills were enhanced (score 3.8/5). 80% of the students believed that the level of difficulty of the Robotics Challenge was just about right, with none of them finding it either too easy or too difficult. Before the Robotics Challenge students did not feel excessively interested in robotics (40% Disinterested or Neutral, 60% Interested), while after the challenge students reported a different level of interest (100% Interested or Extremely Interested). However, student opinions were split 50%-50% on whether there was enough time (2 working days) to complete the task assigned.

The following are some free text comments:

- *"Everything that was taught through all the lectures of the unit, was practically demonstrated in the challenge. It was extremely helpful to understand thoroughly the workings of the sensors, implications when building a robot, different constrains in the map etc."*
- *"The theory helped very much in the overall completion of the challenge"*
- *"It was a very fun process and very helpful"*
- *"A very very positive experience, it was one of the best moments of the semester. Even the mood of the class and the willingness to collaborate and compete at the same time was a very special experience"*
- *"In regards to the time, at the beginning, before starting the challenges it looked as long, but we had such a good time and time flew quickly"*
- *"The things done in the challenge were not of hard complexity, but the outcome of the things learned were of very high importance"*
- *"It was fun and helped us to gain knowledge. Furthermore presenting it to people outside the Department was very interesting"*

6 Conclusions

We have presented our experience in integrating agents and robotics in our Computer Science Curriculum. The concept of intelligent agents is spread throughout the modules at all years of studies. A specialised final year module is central to teaching and learning multi-agent systems and principles of robotics. NetLogo and Lego Mindstorms are used to facilitate hands-on practice with a small learning curve as well as assessment through simulation and realistic tasks, respectively.

We have also presented the Robotics Challenge which allows students to integrate the knowledge they obtained in a number of AI modules, and to practically apply knowledge and skills in order to solve a real problem. By breaking down the challenge into smaller challenges, students are asked to develop robots over a short period of two days without limiting the their ingenuity and inventiveness. The challenge is very well received by students since they have the opportunity to demonstrate their robots outside the University and thus promote Computer Science in general, and agents and robotics in particular.

References

1. Álvarez, A., Larrañaga, M.: Experiences incorporating lego mindstorms robots in the basic programming syllabus: Lessons learned. Journal of Intelligent & Robotic Systems 81(1), 117 (2016)
2. Bloom, B., Krathwohl, D., Masia, B.: Bloom taxonomy of educational objectives. Allyn and Bacon, Boston, MA. Pearson Education (1984)
3. Danahy, E., Wang, E., Brockman, J., Carberry, A., Shapiro, B., Rogers, C.B.: Lego-based robotics in higher education: 15 years of student creativity. International Journal of Advanced Robotic Systems 11(2), 27 (2014)
4. Estrada, F.: Practical robotics in computer science using the lego nxt, an experience report. In: Proceedings of the 22nd Annual Conference on Innovation and Technology in Computer Science Education, Bologna, Italy (2017)
5. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Enhancing NetLogo to simulate BDI communicating agents. In: Hellenic Conference on Artificial Intelligence. pp. 263–275. Springer (2008)
6. Sakellariou, I., Kefalas, P., Stamatopoulou, I.: Teaching intelligent agents using NetLogo. ACM-IFIP IEEIII pp. 209–221 (2008)
7. Stamatopoulou, I., Fasli, M., Kefalas, P.: Introducing AI and IA into a non Computer Science graduate programme. Multi-Agent Systems for Education and Interactive Entertainment: Design, Use and Experience: Design, Use and Experience pp. 89–100 (2010)
8. Wiens, J., Monett, D.: Using BDI-extended NetLogo agents in undergraduate CS research and teaching. In: Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2013)
9. Wilensky, U., Rand, W.: An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. MIT Press (2015)