International Conference on Information and Convergence Technology for Smart Society
Jan. 19-21, 2016 in Ho Chi Minh, Vietnam

# Transferring Teaching Knowledge in Multiagent Systems into Research and Development

Ka Lok Man, Hai-Ning Liang, Yong Yue, Zhao Wang, Eng Gee Lim and Yuxuan Zhao
Department of Computer Science and Software Engineering
Xi'an Jiaotong-Liverpool University, Suzhou, China
{ka.man, haining.liang, yong.yue}@xjtlu.edu.cn
Department of Electrical and Electronic Engineering
Xi'an Jiaotong-Liverpool University, Suzhou, China
{zhao.wang01, enggee.lim}@xjtlu.edu.cn

*Abstract*—**In this paper we present our experiences of using teaching to spark some research projects for students to pursue. The context is multiagent systems—that is, a group of individual agents working together in an automated manner. To be able to work as a single unit, the agents will need to interact with each other to cooperate, coordinate, and negotiate. Our approach has led to a number of student-led projects whose outcomes have been positive and their results were published in international conferences and journals. We present our experiences so that other colleagues can follow some of the ideas we used.**

**Keywords-multiagent systems; a multi robots systems; teaching methodology; active learning; a smart car**

## I. INTRODUCTION

A multiagent system [1] is one that consists of a number of agents, which interact with one-another. In the most general case, agents will be acting on behalf of users with different goals and motivations. To successfully interact with each other, they will require the ability to cooperate, coordinate, and negotiate with each other, much as people would do. In multiagent systems, we address questions such as:

- How can cooperation emerge in societies of self-interested agents?
- What kinds of languages can agents use to communicate?
- How can self-interested agents recognize conflict, and how can they (nevertheless) reach agreement?
- How can autonomous agents coordinate their activities so as to cooperatively achieve goals?

After years of research and development, multiagent systems have become one very important area in computer science. The goal of multiagent systems is to seek methods to build systems that are composed of autonomous agents. These autonomous agents are capable of performing the desired global behaviors without much human intervention.

A classic example of multiagent systems is the "Air Traffic Control" system. When a key air-traffic control system suddenly fails, leaving flights in the vicinity of the airport with no air-traffic control support. Fortunately, autonomous air-traffic control systems in nearby airports recognize the failure of their peer, and cooperate to track and deal with all affected flights. Systems take the initiative when necessary and agents cooperate to solve problems beyond the capabilities of any individual agent.

This paper we present an approach we have followed to transfer our teaching experiences on multiagent systems to spark our research and development initiatives. Section II presents our teaching experience on multiagent systems. Two students' projects in multiagent systems are presented in Section III. Finally, concluding remarks are given in Section IV.

## II. TEACHING OF MULTIAGENT SYSTEMS

As in all leading universities, a module of multiagent systems is included in the Computer Science curriculum of our university, Xi'an Jiaotong-Liverpool University, located in Suzhou, China.

### A. Aims and Fit of Module

The aims and fit of the module of multiagent systems are to:

1. introduce the student to the concept and principles of single agents and multiagent systems, and the main applications for which they are appropriate;
2. introduce the main issues surrounding the design of intelligent agents;
3. introduce the main issues surrounding the design of a multi-agent society; and
4. introduce a contemporary platform for implementing agents and multi-agent systems.

### B. Learning Outcomes

On successful completion of this module, students are expected to:

1. understand the notion of an agent, how agents are distinct from other software paradigms (e.g. objects)

and understand the characteristics of applications that lend themselves to an agent-oriented solution;

2. understand the key issues associated with constructing agents capable of intelligent autonomous action, and the main approaches taken to develop such agents;
3. understand the key issues in designing societies of agents that can effectively cooperate in order to solve problems, including an understanding of the key types of multi-agent interactions possible in such systems; and
4. understand the main application areas of agent-based solutions, and be able to develop a meaningful agent-based system using a contemporary agent development platform.

### C. *Method of Teaching, Learning and Research*

Research and teaching sustain and fuel each other, creating a supportive environment for creativity and innovation. This synergy not only facilitates effective knowledge transfer, but also serves to enrich the teaching and learning experience and deepen the students' learning outcomes. This is the vision our University has adopted, to be become a research-led teaching university. To achieve this vision, research and teaching have equal place and serve to complement one another.

The module is formulated within the framework of research-led teaching. As such, in addition to formal lectures and tutorials, linking teaching and research is a primary task in delivering the module of multiagent systems within the module. Students are strongly encouraged to apply knowledge gained in the module to address real world problems while under the close supervision of and in consultation with faculty and advanced doctoral students.

## III. STUDENTS' PROJECTS IN MULTIAGENT SYSTEMS

This section briefly presents two sample student projects based on implementations of multiagent systems concepts and principles.

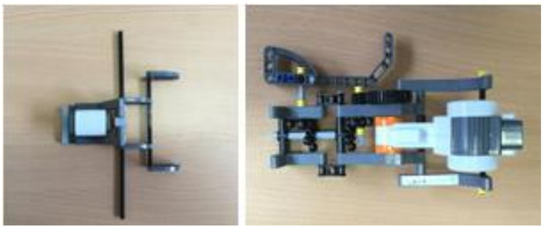
### A. *A Multi-Mobile Robots System*

This multi-mobile robots system contains two types of robots, namely a '*Leader*' and one or more '*Followers*'. In this system, the leader's route is determined and then the follower(s) can plan and modify the routes based on continuously updated feedback from the leader robot. The leader is responsible for path planning and obstacles handling by either making a detour or removing the blockages. According to the system's state and object obstruction in front of the pack, the leader then broadcasts the information to the follower(s) to carry out appropriate actions.

When the lead robot encounters an obstacle, it assesses the situation to see if it is removable and executes the decision to remove the object or take a detour, if the object cannot be remobed. The task of object judgment and removal can be completed by the cooperation between the collision detection sensor and the obstacle removal attachment, and in a case where the removal task fails, the system re-evaluates the plan and redirects the route.

The follower does not have the functions and abilities of distinguishing and handling obstacles and cannot make a decision on how to deal with the obstacle on the route by itself; it must rely on the commands and instructions from leader. This design decision is there so that there is no conflict between the leader and the followers, because if there are more than one follower and every one of them can issue commands, then there would be uncertainty and coordination can become chaotic.

The prototype is implemented on the Java programmable Lego Mindstrom NXT kit [2]. The NXT contains three types of sensors in its package; they are touch sensor, light sensor and ultrasonic sensor, while the servo motors provide mobility. Through the combination of the various components we are able to build a collision detector and obstacle removal attachment device as shown in Fig. 1a and Fig. 1b respectively.



a. b.

Fig. 1. Attachment devices: a) Collision detector and b) Obstacle remover

In this project the LeJOS [3] firmware was used to replace the default LEGO NXT firmware in order to allow us to develop Java based programs, giving us more freedom and power, as opposed to the limited programming environment of LEGO NXT-G.

Several experiments presented in [4] show that this multi-mobile robots system is able to distinguish the types of obstacle robot can encounter and take sensible actions to handle them. Furthermore, robots can interact with each other to guarantee that every robot in the system moves on the desired route.

### B. *Control of an Arduino-based Smart Car*

A user, an Android smartphone, an Arduino-based car and a personal computer are involved in this project. With assistance of the Arduino integrated development environment (IDE) in the PC, sketches are compiled and uploaded into the Arduino board via a USB transmission line. The car and mobile phone are linked via wireless communication. By touching or pressing on the screen or user interface (UI) of an Android phone, a manipulator can send commands to the Arduino microcontroller on the car through Bluetooth and observe corresponding executions accomplished by actuators, like motors.

Components listed in the phone yield data assisting determinations of the command message that will sent to the Arduino and measurement data produced by different sensors equipped on the car are also displayed on the smartphone screen and stored in the server for analyses with the help of the Bluetooth transmitter and Android operating system, which indicates duplex communication. In addition, by uploading new sketches, the user is able to modify the programming of the microprocessor, thereby customizing the performance of the car.

Two gear motors, two wheels, a universal wheel, a battery holder, two 18650 lithium batteries, a switch and two baseboards compose the chassis of the car as shown in Fig. 2. The Uno board, where a motor driver shield and extension shield are plugged in is fixed on this chassis.

Fig. 2. Hardware realization process

Four main functions are designed in this Android application. The fundamental one allows for controlling the cars movement forward, left, right, reverse and stop action based on touching arrows or sliding a white ball as shown in Fig. 3.

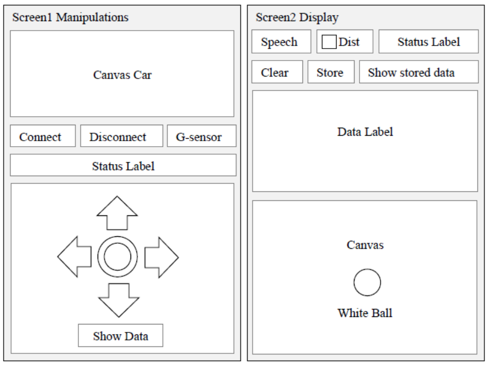



Fig. 3. User interface design

Two modes have been designed in this project. The first mode is wireless control and the second mode is for obstacle avoidance. When the car is operated in mode 1, the only method for controlling the car is by the operations on the smartphone via Bluetooth communication. Four functions are achieved here. The fundamental ones are forward, left, right and reverse movements as well as a stop action based on the touching of arrows or sliding a white ball as shown in Fig. 4. Another form of control is assisted by the gravitational sensor which refers to the accelerometer sensor built-in to the Android smartphone. The third one is to display, store, show and clear measurements sent by Arduino microcontroller. Finally, an essential voice control based on Google Voice Search that can handle receiving data and sending some instruction messages is also included.

If mode 2 is performed, the car keeps going forward until an obstacle appears within the defined threshold distance. After exploring the barrier, it will stop and detect distances in the front-left and front-right directions. If both of them are smaller than a defined value, the car moves backward for a while and measures these two distances again. If there is more space in the front-left direction, it will turn around to its left side by around 90 degrees; otherwise, it turns around to the right side by around 90 degrees. The loop is then is performed continually. A detailed account of the experimental results is given in [5].

## IV. SUMMARY

In this paper we presented our experiences to use classroom teaching on multiagent systems to frame research projects for students. Our experiences have been quite positive, as can be seen from the two sample projects we presented in the paper. We based our delivery of the module in trying to implement the vision of our university to become a research-led teaching university. In this vision, teaching and research go in parallel, and we are encouraged to not only present the latest research findings within a particular area but to also involve students to do projects as part of the requirements of any taught modules. As our case shows, research-led teaching can take place even for undergraduate modules, an approach that can be implemented elsewhere.

## REFERENCES

[1] Wooldridge, Michael (2002). An Introduction to MultiAgent Systems. John Wiley & Sons. ISBN 0-471-49691-X.

[2] “Home - Mindstorms LEGO.com.” [Online]. Available: http://www.lego.com/en-us/mindstorms/. [Accessed: 21-Jul-2015].

[3] “leJOS,” Wikipedia, the free encyclopedia. 27-Jun-2015.

[4] Yuxuan Zhao, Ka Lok Man, Hai-Ning Liang, Wei Wang and Yong Yue and Taikyeong Jeong, Design of Intelligent Algorithms for Multi-Mobile Robot Systems, International SoC Design Conference, Korea, 2015.

[5] Zhao Wang, Eng Gee Lim, Weiwei Wang, Mark Leach, Ka Lok Man, Vladimir Hananov, Eugenia Litvinova, Svetlana Chumachenko and Vladimir Miz, Design of an Arduino-based Smart Car, International SoC Design Conference, Korea, 2014.