Running Experiments on DipGame Testbed (Demonstration)

Angela Fabregues, Santiago Biec and Carles Sierra
Artificial Intelligence Research Institute (IIIA-CSIC)
Campus Universitat Autònoma de Barcelona, 08193 Bellaterra, Catalonia, Spain
{fabregues, sierra}@iiia.csic.es, santiago@dipgame.org

ABSTRACT

DipGame is a testbed for MAS negotiation involving humans. It is very appropriate to run experiments that mix humans and agents. In this demonstration we introduce an application to facilitate the execution of experiments on several machines and with a friendly graphical user interface.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation

Keywords

application, negotiation, testbed, diplomacy game.

1. INTRODUCTION

There are recognised difficulties of running experiments on negotiation involving both human agents and software agents [3]. These difficulties are delaying the production of automated negotiation agents. First, most research work on negotiation techniques does not consider humans as counterparts of automated negotiation agents. Second, enticing humans to participate in negotiation experiments is difficult because the negotiation environment is artificial and not attractive, and because the language to use in interactions is unnaturally constrained. Some of the barriers of the latter type are solved by the DipGame testbed [2].

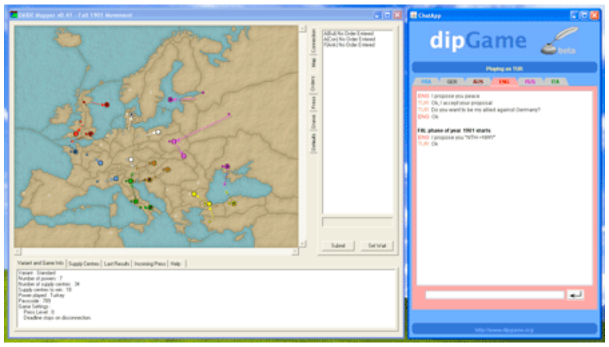
DipGame provides an environment where agents incarnate one of the seven Great European Powers as defined by the Diplomacy Game (`http://en.wikibooks.org/wiki/Diplomacy/Rules`). This game temporally splits in turns where all players move the units that they control over a map of Europe. The goal of the game is to conquer Europe and this is achieved performing cooperative moves with other players that where agreed in the negotiation round that takes place before each turn is executed. Those agreements, usually alliances, can be dishonoured. All the game is about understanding the relationships among agents, knowing to what extend we can ask for help and guessing whether someone is trustful or not [4].

In DipGame we cope with the language problem by providing a formal language and a library that translates human messages about Diplomacy from (a restricted set of) English into the formal language, [2]. This library and a friendly user interface to write the messages are integrated as a web application that everybody can use to play online against other humans and software agents. This interface is available at `http://www.dipgame.org` and helps attracting Diplomacy players to take part in our experiments.

The analysis of the data produced by negotiation experiments, consisting of several game executions, is made with the help of DipTools [1]. This tool allows the experimenter to group the results of sets of game executions in order to compare and analyse them in a intuitive and graphical way.

Several research labs have shown their interest on using the DipGame testbed and have started to design negotiating agents. Building a DipGame negotiation agent became an assignment for some undergraduate and master students who reported the difficulty of testing their agents offline without a simple graphical interface that would allow to set the experimental variables and to collect the resulting experimental data. Offline agent testing is crucial to ensure that software agents perform well and do not crash while playing a game with humans, as this would demotivate them to continue in the experiment. This is the main goal of the experiment manager that we will present in this demo: to facilitate the offline testing of DipGame agents.

Section 2 describes the software that has been developed and Section 3 provides an example of how to use the experiment manager.



ENG I propose you peace
TUR Ok, I accept your proposal
TUR Do you want to be my allied against Germany?
ENG Ok

Fall phase of year 1901 starts
ENG I propose you "NTH->NWY"
TUR Ok

Figure 1: Screenshot of the human graphical interface, ChatApp. The chat is on the right and the map on the left.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.
Copyright © 2012, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

2. SOFTWARE DESCRIPTION

The experiment manager, released as the DipGame GameManager, provides a graphical user interface where games can be set up. The application is multi-platform and runs on several devices. It allows to select the players that will take part in an experiment. Each player can be either a provided software agent (currently four such agents are included in the software release), a software agent programmed by the experimenter, or a human interacting through a graphical interface.

The graphical interface for humans is called ChatApp and can be seen in Figure 1. This human interface is included in GameManager and it is also released as an standalone application. ChatApp provides a chat functionality similar to most instant messaging software available in the market. In addition, this chat translates the natural language messages into the formal language that automated agents understand and vice-versa. This translation is done in such a way that players do not know whether the opponent is a human or an agent. Finally, ChatApp renders a map used to select the movements to perform.

The manager allows to set a player to *empty*. This means that the game can be launched even though there are players missing. Once launched, the game will wait for those missing players to connect using the IP address and port indicated by the manager. Missing players can be standalone applications. ChatApp is an example of such standalone application. Software agents can also be executed this way. In section 3 we present an example of experiment concurrently executed over several machines.

3. EXAMPLE

The typical users of the experiment manager are researchers that are developing their own software agents. To run an experiment you have to first download the tool and incorporate your agent into the manager. The software and the instructions for incorporating agents are available at the GameManager section of the DipGame site. Next, you can run the manager and set the experiment selecting the players you like to take part in it. Among the available players you will find those software agents that you incorporated. Finally, run the experiment and save the results into a file.

In [1] we described how to analyse the results of an experiment involving several game executions with negotiating and non-negotiating agents. To run an experiment involving, for instance, four copies of your software agent (each one with possibly different parameter values) and three human agents, you will need at least three machines to interface with the three humans. Three machines would be enough as one might run the experiment manager and the other two the ChatApp standalone application. Thus the experiment manager would have four players set to the software agent, one player set to human and the last two players set to *empty*. When running the experiment, the manager launches the game with two players missing and launches also a ChatApp integrated with the manager. This human interface can be used by one of the humans. For the game to be able to start, the other two humans should connect to the manager by introducing vis their graphical interface the IP of the manager that is shown in the manager main window.

The human interface integrated with the experiment manager should be used only for testing purposes as the human using it would have access to private messages sent between the other players. When the game ends, or when the game is cancelled,[1] the results can be stored in a file. Then, we can take one or several experiment result files, upload them into DipTools and visualise the results as shown in Figure 2 for a particular experiment.

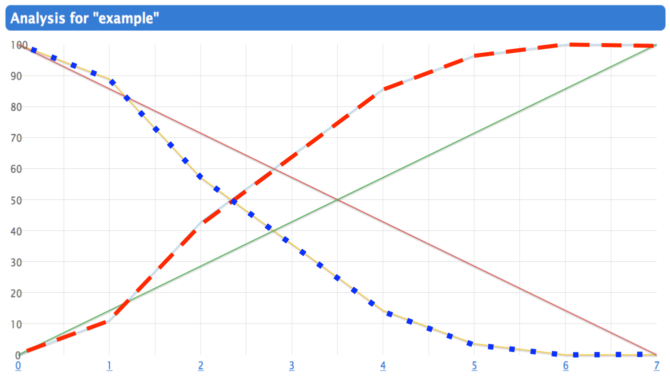


Figure 2: Example of chart extracted from [1]. Percentage of games won per number of negotiating agents. The dashed line represents the percentage of victories of negotiating agents and the doted line the percentage of victories of non negotiating agents. The continuous lines (increasing and decreasing) represent the expected percentage of negotiating and non-negotiating agents in case they all were equal. This particular graphic shows that negotiating agents perform better in the experiment.

This paper is accompanied with a video demonstration available at `http://www.dipgame.org/media/AAMAS2012demo`.

4. ACKNOWLEDGMENTS

Research supported by the Agreement Technologies CONSOLIDER project under contract CSD2007-0022 and INGENIO 2010, by the Agreement Technologies COST Action, IC0801, and by the Generalitat de Catalunya under the grant 2009-SGR-1434. This work is supported under project CBIT, which is funded by Spain's Ministry of Science and Innovation under grant number TIN2010-16306.

5. REFERENCES

[1] A. Fabregues, D. López-Paz, and C. Sierra. Diptools: Experimental data visualization tool for the dipgame testbed (demonstration). pages 1315–1316, Taipei, Taiwan, 02/05/2011 2011.

[2] A. Fabregues and C. Sierra. Dipgame: a challenging negotiation testbed. *Journal of Engineering Applications of Artificial Intelligence*, 24:1137–1146, 10/2011 2011.

[3] R. Lin and S. Kraus. From research to practice: Automated negotiations with people. http://u.cs.biu.ac.il/ linraz/Papers/linetal-practice.pdf.

[4] R. Sharp. *The Game of Diplomacy*. 1978. http://www.diplom.org/ diparch/god.htm.

[1]The experiment execution can be canceled at any time from the experiment manager window showing the real time results.