University of Auckland

COMPSCI 765

Interactive Cognitive Systems

Multi-Agent Mafia

Author:	*UPI:*
Bradley Moorfield	bmoo063
Sebastian Cram	scra069
Derek Galea	dgal061

June 3, 2015

Contents

1 Introduction

1.1 Overview

Mafia [1] is a social party game where an informed minority, the mafia, conflicts with an uninformed majority, the innocents. The game is typically played with seven to ten players, with two to three mafia for best results. Many variations on the rules exist, but the basic game focuses on the village as a whole debating the identities of the mafia and voting on one player to lynch during the day, with the mafia choosing one player to kill during the night. The game ends when all the mafia have been eliminated, or the mafia outnumber the innocents. Outside of these rules, the players may do anything to try convince the other players of their innocence, or accuse another player of being mafia.

Other roles can be assigned, including a medic and a sheriff. The medic has an opportunity during the night to select one individual to be saved, should they be targeted by the mafia. The sheriff has an opportunity during the night to select one individual and be informed by the game master whether that person is mafia or not.

The game master controls the flow of the game and decides when to advance the game onto the next phase. They request the individuals with a particular role to take action at the appropriate time and announce the results of the nights activities. The game master simply applies the rules of the game, so their actions will be coded in as rules based on the game state.

The other players will not have full knowledge of the game state, so they will form beliefs about the other agents playing the game. They will have beliefs about their own and others roles in the game. They can observe the other agents actions in the game and form beliefs based on those observations. Some basic actions include making an accusation, asking others about their beliefs and making claims about their role in the game. To give them some way to assess the truthfulness of other agents claims we may include unintentional actions like a wavering voice, hesitation or perspiration. So there would be same chance if an agent lies about their role, they do so in a wavering voice. For example, one agent accuses another of being mafia. The accused claims they are the medic and perspires. The accuser then forms the belief that they are mafia, as do the other agents. Then when the vote is called they are likely to lynch them.

One potential way to make the inferences about agents roles more complex is to give the agents some rudimentary personality. Then if the accuser believes the accused is a shy person there is a competing explanation for their perspiration. Other relevant personality traits would be aggressiveness and honesty. These traits would influence how often they make accusations or lie.

1.2 Motivation

Mafia allows for a variety of challenges from mathematical, computational, and cognitive standpoints. Braverman et al. [2] shows extensive mathematical modelling can be applied to even a simple environment. Computationally it follows from from the likes of the Facade and Versu systems in having a simulated set of agents interacting with a user in order to tell a story. The simulated agents also need to interact with each other, as each agent is independent.

Haan et al. [3] have previously implemented a simple mafia playing agent to demonstrate their multi-agent framework, however little reasoning of other agents true beliefs was performed. Our goal is for agents (both real and simulated) to be able to make inferences as to the true status of other agents based on individual announcements and other observables, and perhaps uniquely it requires agents to lie and identify the lies of others. This is a cognitively demanding task, so it will be interesting to see if we can find ways to make the agents intelligently deceptive.

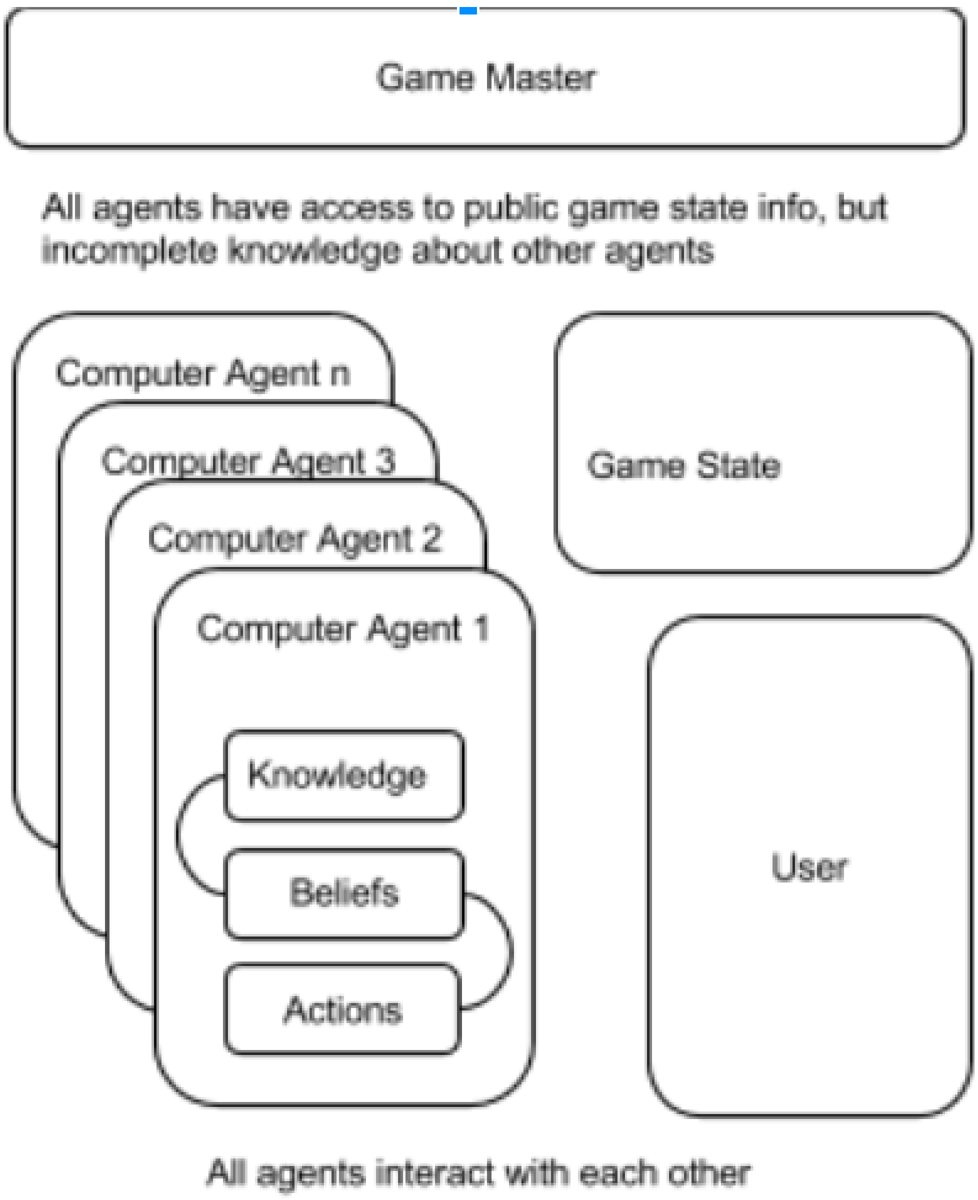
Game Master
All agents have access to public game state info, but incomplete knowledge about other agents
Computer Agent n
Computer Agent 3
Computer Agent 2
Computer Agent 1
Knowledge
Beliefs
Actions
Game State
User
All agents interact with each other

Figure 1: Entity diagram for Multi-Agent Mafia system

2 Analysis

The game starts with the game master assigning roles randomly to all of the players. The game master initiates a discussion phase, then begins the first night phase. The game master first asks the mafia to decide together who to assassinate. Then the medic is called on to select someone to save. Then the sheriff is called on to investigate one player. The game master announces the results of the night and assesses whether the game end conditions are met. Agents update their knowledge and beliefs based on the results. The game master then moves the game into the discussion phase. Agents can make announcements about their beliefs or ask others about their beliefs. To ensure discussions dont go on forever we can limit each agent to two speech actions (this is a problem in real life games and the game master has to exercise judgement about when to terminate discussions). The game master then moves the game into the accusation phase. Each agent can accuse up to two other people. The game master then allows the accused to defend themselves. They may make claims about themselves or a belief about another player. For example, I am the sheriff, and I know Bob is mafia or just deny the accusation or remain silent. Then each player casts their vote publically.

From information about who voted to lynch, inferences can be made, so knowledge about how people voted should be added to each agents knowledge base. For example, if one player made an accusation and claimed they were the sheriff and knew the accused was mafia, a vote is taken and the accused is lynched and they turned out to only be a citizen, the accuser was lying. This may lead players to conclude the accuser is mafia.

The game master again assesses whether the game end conditions are met. If not, another round of discussion takes place. Then another night falls and the night cycle repeats until one team has won.

3 Design

3.1 Overall Architecture

Our system is implemented using the Jason Multi-Agent System [4], which provides a framework that allows multiple agents to run simultaneously using a logic-programming BDI language called AgentSpeak [5], communicate with each other, and modify the underlying architecture of agents and the environment.

AgentSpeak encodes agents as a set of beliefs and plans. Beliefs are first-order predicates that define the knowledge of the world as the agent knows it. Plans consist of an event trigger (either the addition or deletion of a belief), a context defined as the conjugation of beliefs, and a sequence of actions that the agent has to achieve when a plan is chosen for execution.

The extended implementation of AgentSpeak used in Jason provides two important internal actions: broadcast and sent. These actions allow an agent to either broadcast a message to all other agents, or send a message to a specific agent. Messages consist of an illocutionary force, a sender, a reciever, and a belief literal. Agents communicate their beliefs by broadcasting their accusations and denials to all, or participate in the game by sending their votes to the game controller.

The overall structure of the system is shown in figure 2. The Jason interpreter is responsible for distributing messages from a broadcast or send, so all agents interact with each other through this central hub. In this implementation, the game controller is implemented as an agent that does no reasoning, it simply acts as a man in the middle to intercept communication and keep track of the game.

The game controller has a modified agent architecture. This is a feature of Jason whereby we can write Java code that changes how the agent deals with receiving beliefs, in this case it keeps track of the state of the game, and counts votes that are sent by other agents. The game controller is also able to send out beliefs that tell the agents when the state of the game changes, this happens when the user presses the 'next state' button on the interface presented by the game controller.

The final piece of the puzzle, the user, interacts through the game controller interface. The interface consists of buttons that allow the user to perform actions similar to those that the agents can perform, such as accusing, denying, voting during the town vote or voting during the mafia decision. The buttons become enabled and disabled depending on the state of the game and whether or not the human agent is still alive. The game controller broadcasts messages on behalf of the user whenever the user presses the button for a corresponding action.

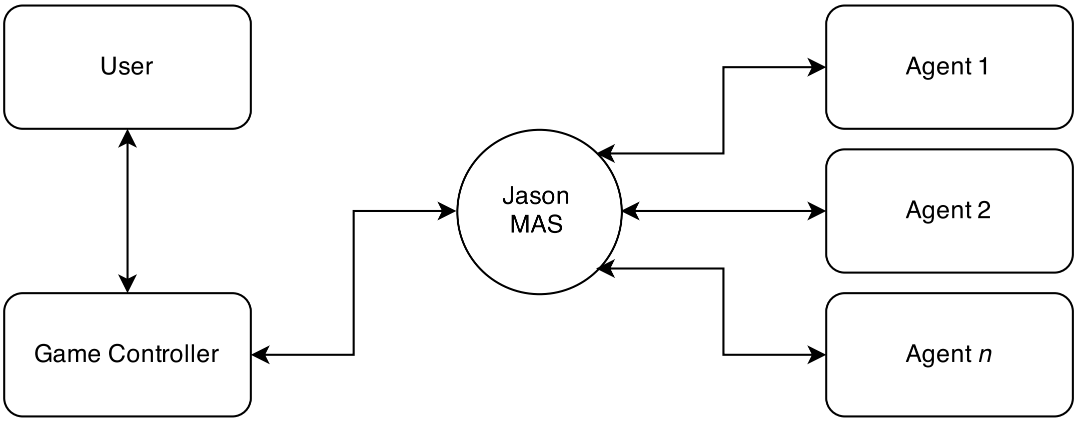

Figure 2: Overall architecture of the Multi-Agent Mafia system

3.2 User Modelling

3.3 Planning Logic

4 Practical

4.1 Ilustration

4.2 Limitations

There are four key components which could further be implemented. Those being roles, personality, memory, and discourse.

4.2.1 Roles

We only implemented the simplest roles required to form a coherent game, those being villagers and mafia. There are other roles that could be assigned however. These include the doctor who, after the mafia has selected a victim, is allowed to choose one person to live through the night, and the sheriff who once each night can consult the system for the role of another agent.

4.2.2 Personality

Each agent within the system is presently homogeneous. With further development we could see the agents gaining their own seperate attribtes and personalities. For instance one agent could be partiularly charismatic allowing other agents to be more easily convinced by their accusations. Two agents could perhaps feud, leading them to often accuse each othe regardless of plausability.

4.2.3 Memory

At present each game takes place in isolation. Further implementation would perhaps allow for a continued memory for the agents. Allowing for grudges that last between games, and forming opinions on the honesty and charisma of other agents.

4.2.4 Discourse

In it's simplest form we implemented only the ability to 'accuse' and 'deny' membership of the mafia. While a full natural language interface might be achievable, we would focus more on expanding how each of those actions take place. Accusations through inference "agents 1 accused agents 5, now agent 1 is dead" vs passion "agent 3 was mafia last game and they killed me!". Even during the voting stages perhaps being able to defend others to sway certain votes.

4.3 Demonstration

A working run playing as a member of the mafia is available here

5 About the Authors

5.1 Bradley Moorfield

Initial planning and development. Program design. Coding. Report copy.

5.2 Sebastian Cram

Initial planning and development. Program design. Report copy. Report formatting

5.3 Derek Galea

Initial planning and development. Program design.

6 Conclusion

References

[1] The Graduate Mafia Brotherhood of Princeton University *http://www.princeton.edu/ sucharit/ mafia/* Accessed March (2015)

[2] Braverman, M., Etesami, O., Mossel, E. (2008). Mafia: A Theoretical Study of Players and Coalitions in a Partial Information Environment. Annals of Applied Probability.

[3] Haan, H. W. D., Hesselink, W. H., Renardel de Lavalette, G. R. (2004). An abstract multi-agent framework applied to a social interaction game. University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

[4] Jason *http://sourceforge.net/projects/jason/* Accessed April (2015)

[5] Rao, A. (1996). AgentSpeak(:): BDI agents speak out in a logical computable language. Lecture Notes in Artificial Intelligence.