

Team Formation in Agent-Based Computer Games

Hongen Lu

Department of Computer Science and Computer Engineering

La Trobe University

Bundoora, Melbourne

VIC 3086, AUSTRALIA

helu@cs.latrobe.edu.au

ABSTRACT

Intelligent agents play an important role in computer games. Research on intelligent agent architecture, knowledge representation, goal-directed behavior are all directly relevant to improving the intelligent agents in computer games [9]. Agents are suitable to model and develop characters in computer games. To form an agent team in a computer game environment is extremely difficult. In this paper, a Naive Bayesian model is proposed for agent based characters to form teams. This approach brings a new angle for us to solve the connection problem for agents to work collaboratively in a dynamic and open environment, such as computer games.

1. INTRODUCTION

Intelligent agents play an important role in computer games. Research on intelligent agent architecture, knowledge representation, goal-directed behavior are all directly relevant to improving the intelligent agents in computer games [9]. Agents are suitable to model and develop characters in computer games. In many developers point of view, a character in a computer game is an intelligent agent with plans and goals. The number of networked computer games, such as ALICE Chat Bot and Fantasy XI, increases dramatically in recent years. In such an open network environment, to form agent cooperation in a computer game environment is extremely difficult.

Research on intelligent agent cooperation provides solutions on this problem. For example, SportsAgents [7] system a mediator-based multi-agent open system. In this system each information agent has the ability to communicate with others, using the commonly agreed KQML language; they can exchange messages about their internal knowledge and capabilities despite that having different internal implementations. And more importantly, mediator in this open system serves as a middle man to introducing agents to each other, and guides them into cooperation. This is

achieved in a mediator by combining matchmaking and cooperation forming together. In this paper, multiple agent service matchmaking strategies are described, and a Naive Bayesian model is given to decide the agent cooperation team for mediator to recommend to require agents. Combining matchmaking and cooperation forming gives a new angle for agents to collaborate. Traditionally, they are two separate processes [6]. In order to let agents form a cooperation according to the relations and semantics of their capabilities, it is necessary to combine the two together.

2. AGENT BASED CHARACTER ROLES

First of all, what is the function of a character in a computer game? and how to describe the capability an agent can offer? To answer these questions, let us look at the following definition.

DEFINITION 1. (Game Character Role) *A role in a computer game, $\mathcal{R} = \langle \mathcal{N}, \mathcal{I}, \mathcal{O}, \mathcal{F} \rangle$, is a character provided by an agent, where \mathcal{N} is the name of this character; \mathcal{I} , \mathcal{O} are the input and output; and \mathcal{F} is the functionality \mathcal{R} can achieve, i.e. $\mathcal{F} : \mathcal{I} \rightarrow \mathcal{O}$.*

For a role \mathcal{R} , there are features associated with it, such as constraints for input and output constraints respectively; reliability and quality are other two important features. Different application domain may concern different role features.

2.1 Role Relationships

Different characters in a computer game have different functions and roles. Among them there exist relationships. These relations are characterized below. Let R_i denotes the role of a game agent \mathcal{GA}_i .

- **Identical Role:** $R_1 = R_2$. This means the two game agents can provide the same function in spite of the fact that they may have different role names. Obviously, two identical roles can substitute each other.
- **Sub Role:** $R_1 \subset R$. This relationship characterizes two services offered by agents, in which one role's function is only a part of another. For instance, in a fighting game a game agent can only use rifle to shoot. This role is a sub role of an agent who is capable to fire multiple weapons.

- **Substitute Role:** a role R_1 can be substituted by role R_2 , $R_1 \leftrightarrow R_2$. From the above description, we know that identical role and sub role are two special cases of substitute role relationship. But the difference is that identical roles can substitute each other, while the sub role can only be alternated by its super role, not vice versa.
- **Partial Substitute Role:** $R_1 \cap R_2 \neq \phi$. This relationship describes two services that have some common sub roles. In some circumstances, partial substitute roles can be alternated by each other, such as where the role agent is offering, just by chance, the common sub role with its partial substitute role; that is, the agent is not offering its full function to others at the moment.
- **Reciprocal role:** $\exists R = (R_1 \cup R_2) \text{ AND } (R_1 \cap R_2) = \phi$, then R_1 and R_2 are reciprocal with R . If two services are reciprocal, that means they have no sub roles in common, but they can work together to offer a “*bigger*” role. From this definition we know that in case there is no current agent available to provide the “*bigger*” role, these two reciprocal services can cooperate as a single agent for this task. This gives us a message that by combining the current agents in a different manner, we can tailor the system to meet new requirements.

To find the role relationship between two game agents, the middle agent should keep the knowledge of the domain role. The knowledge the middle agent would need to match the functions of game agents is effectively an ontology of services. The agent role ontology contains all the roles of game agents as well as their relationships. Agent role ontology is a key element to perform meaningful services description and matching.

2.2 Multiple Matchmaking Strategies

Given the above role relationships, a matchmaking process is needed to find the right game agent, or game agents, to cooperate in order to fulfill a role. Considering various features of a role, a number of matchmaking strategies are described below.

- **IO Match** This is the simplest strategy that only matches the types in the input and output fields of role advertisements against the correspondent field in requirements. It makes sure that a provider can take the inputs of requester, and its outputs are compatible with the requester’s.
- **Constraint Match** Constraint matchmaking considers the constraint parts of two agent services.
- **Exact Match** Exact match is most strict matchmaking. It requires both the types and constraint fields are well matched. This strategy deals with the services that have the same functions but with different variable and type names. Considering the huge amount of Web-based applications which implemented over times and locations, there are many cases that developers may select different naming space.

- **Partial Match** Partial match is a combination of type match and constraint match, but both loose a little bit. This strategy aims at services that are not completely matched, but have some functions in common.
- **Cooperative Match** Cooperative matchmaking is a process based on a collaborative partnership between game agents. In our SportsAgents system, a mediator is introduced to solve the connection problem. SportsAgents is an open multi-agent system to answer a user’s query about sports, for instance “which is the best sports city in Australia?”. In SportsAgents three types of agents, mediator, game agents and interface agents, are implemented. To find the best sports city may require the cooperation of different sports information agents. The mediator finds the current available game agents who have the capability that the query agent (information consumer) is asking for. In case no available agent can fulfill the query role itself, the mediator will infer the available services to find a set of available game agents that can cooperate in some way to provide the requested role.

3. AGENT TEAM FORMATION

Cooperation is often presented as one of the key concepts that differentiates multi-agent systems from other related disciplines such as distributed computing, object-oriented systems, and expert systems [2]. However, there is not a precise definition for agent cooperation. Generally, we believe the fact or decision that different agents work together, intentionally or unintentionally, to solve a problem, is an example of cooperation. All the agents involved in a cooperation form a team. From a single agent’s stand point, he may know he is only doing his job rather than “*realizing*” he is joining cooperation with other agents, but from the system’s view, we observe cooperative actions among agents. Coalitions or cooperation of agents can work more effectively than individual agents in many multi-agent settings. Determining which coalition should form, or what agents to work together with, is a difficult problem.

Ketchpel [4, 5] proposes a fully distributed mechanism for exchanging labor among the agents to exploit the different abilities of the agents. His work formalists the cooperation formation problem in decision theoretic and game theoretic terms. The method is based on the Contract Net [1] to distributively and efficiently determine coalitions that will be approximately “stable”. Other market-oriented approaches, such as in [10], match producers and consumers of services, but allow only well specified transfers of goods among them, falling short of the notion of forming a collaborative group.

4. BAYESIAN NETWORK

A Bayesian network is a directed, acyclic graph that compactly represents a probability distribution. In such a graph, each random variable is denoted by a node. A directed edge between two nodes indicates a probabilistic influence from the variable denoted by the parent node to that of the child. For many problems, it is possible and even straightforward to construct a Bayesian network from prior knowledge alone [8]. Consequently, over the last two decades, the Bayesian network has become a popular representation for encoding uncertain knowledge in expert systems [3].

5. TEAM FORMATION MODEL

In this section the definitions of simple team and composite team are given; and a Bayesian model is given for agent cooperation formation. This model will help game agents to decide which cooperation to join in. Given a set of game agents with different abilities and different information, there may be many opportunities for cooperation among the agents. Different cooperation among game agents results differently in term of the utility of a single agent and the final result of the game.

5.1 Simple Team and Composite Team

In the real world, the functions agents offered vary from each other. Sometimes in an open system, the role you require can be simply provided by an agent; or you have to discover a set of agents to perform your task; however sometimes it is not the case; you can not find the role you are looking for. From this point of view, we can classify agent teams into two categories: simple team and composite team.

DEFINITION 2. (Simple Team) *A team consists of one agent is defined as a simple team.*

It would be easy for us to use a role identifier in a description to present the team. We can not expect that all kinds of roles we request can be simply provided by a single agent. It is neither practical, nor possible sometimes to build so many agents for various tasks.

DEFINITION 3. (Composite Team) *A composite team is a role that is achieved by a set of game agents working cooperatively. We use these game agents' role names and their cooperation relationships to denote the composite team.*

For example, a composite team $CT = AND(R_1, OR(R_2, R_3), R_4)$ denotes the role that three game agents $\mathcal{GA}_1, \mathcal{GA}_4$, and either of \mathcal{GA}_2 or \mathcal{GA}_3 , work together to provide. From the above definition, it is noted that the result of our cooperative matchmaking strategy is a set of composite teams.

Please note, in an open system, due to the dynamic change of its components, a single team may become a composite team or vice versa. This is because in such an environment, game agents can join in and quit at any time; so the number and functions of roles in a computer game are also changing time by time.

5.2 Team Formation Decision Model

Following the decision making process is given when a game agent gets a set of composite teams from the mediator; and a Naive Bayesian model is employed to choose a better composite team for the agent to join.

Suppose D is the decision to make, $D_k \in \{Accept, Reject\}$. Given a set of composite teams CT_1, \dots, CT_n , for each composite team CT_i we have p_i participating game agents $\mathcal{PGA}_i = \{\mathcal{GA}_{i1}, \dots, \mathcal{GA}_{ip_i}\}$, and for each game agent \mathcal{GA}_{ij} we select m features as $F(\mathcal{GA}_{ij}) = \{F_{ij}^1, \dots, F_{ij}^m\}$.

From the Bayes theorem, we know the probability to make decision D_k under an instance ct of composite team is:

$$P(D = D_k | CT = ct) = \frac{P(CT = ct | D = D_k)P(D = D_k)}{P(CT = ct)} \quad (1)$$

As we know each feature F_{ij}^h , $h = 1, \dots, m$, is conditionally independent of every other features in most of agent based computer games. This is the most restrictive assumption embodied in the Naive Bayesian model. Formally this yields:

$$P(CT = ct | D = D_k) = \prod_{i=1}^n P(CT = ct_i | D = D_k) \quad (2)$$

For each $P(CT = ct_i | D = D_k)$, we have

$$P(CT = ct_i | D = D_k) = \frac{\sum_{j=1}^{|\mathcal{GA}_i|} \lambda \sum_{h=1}^m P(F_{ij}^h | D = D_k)}{m \times |\mathcal{GA}_i|} \quad (3)$$

where

$$\lambda = \begin{cases} 1 & \text{if } \mathcal{GA}_{ij} \text{ participates AND in } CT_i \\ 1/q & \text{if } \mathcal{GA}_{ij} \text{ participates OR in } CT_i \text{ with } q \text{ } \mathcal{GA}s \end{cases}$$

Using the above model a game agent can evaluate the result it gets from the mediator and choose a set of promising role providers whose probability of $P(Accept | CT = ct)$ is the highest to collaborate with. This process will improve the performance of a single game agent.

6. AN EXAMPLE

Suppose there are three composite teams available: CT_1 , CT_2 , and CT_3 , where $CT_1 = AND(GA_1, GA_2)$, $CT_2 = AND(GA_1, GA_3, GA_4)$, and $CT_3 = AND(GA_1, OR(GA_2, GA_3), GA_4)$. In this case, composite team CT_1 has two participating game agents \mathcal{GA}_1 and \mathcal{GA}_2 , and they are working together to fill the role of CT_1 . For CT_2 and CT_3 , they have 3 and 4 participating game agents respectively.

For each game agent, the two features of reliability and quality are selected. Reliability is used to measure the percentage a game agent can reply for the requests sent to it; and quality measures how many answers the game agent gives are satisfactory among all the replies for a certain request.

	Reliability		Quality	
\mathcal{GA}_1	.9	.4	.6	.3
\mathcal{GA}_2	.8	.4	.7	.3
\mathcal{GA}_3	.7	.3	.6	.4
\mathcal{GA}_4	.6	.5	.6	.4

Table 1: Participating Game Agents Features

According to the values in Table 1, the probabilities for each composite team can be calculated. Using Equation 3, we have:

$$\begin{aligned} P(CT_1 | Accept) &= 0.75 \\ P(CT_2 | Accept) &= 0.5 \\ P(CT_3 | Accept) &= 0.38 \end{aligned}$$

Also based on experiments, we have $P(Accept) = 0.88$, $P(CT_1) = 0.85$, $P(CT_2) = 0.8$, and $P(CT_3) = 0.75$. After applying Equation 1, we have:

$$\begin{aligned} P(Accept|CT_1) &= 0.78 \\ P(Accept|CT_2) &= 0.55 \\ P(Accept|CT_3) &= 0.45 \end{aligned}$$

From the above calculation, we can make our decision to select the composite team CT_1 , that is, giving the task to participating game agents \mathcal{GA}_1 and \mathcal{GA}_2 .

7. CONCLUSION

Research on intelligent agent cooperation provides solutions on team formation in computer games. In this paper a method to combine game agent role matchmaking and team formation is proposed. For flexible and semantic matchmaking, multiple strategies are provided; and based on the results of matchmaking, a Naive Bayesian decision model is given for agents to form computer game teams. This approach, which considers matchmaking and team formation as one process, brings a new angle to solve the connection problem for agents to work collaboratively in a dynamic and open environment, such as the networked computer games. Because in an open system, it is necessary to have a method for agents to form a coalition to solve problems collaboratively. A Bayesian based model for agent cooperation formation is presented, which will help agents to decide the team. This model is useful in networked computer games.

8. REFERENCES

- [1] R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. In A. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 333–356. Morgan Kaufmann, 1988. San Mateo, California.
- [2] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman. On cooperation in multi-agent systems. *The Knowledge Engineering Review*, 12(3):309–314, 1997.
- [3] D. Heckerman, D. Geiger, and D. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
- [4] S. P. Ketchpel. Forming coalitions in the face of uncertain rewards. In *Proceedings of National Conference on Artificial Intelligence (AAAI-94)*. AAAI Press, 1994.
- [5] S. P. Ketchpel. Coalition formation among autonomous agents. *Lecture Notes in Computer Science*, 957, 1995.
- [6] S. Kraus, O. Shehory, and G. Taase. Coalition formation with uncertain heterogeneous information. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS03)*, pages 1–8, Melbourne, Australia, July 2003.
- [7] H. Lu and L. Sterling. Sportsagents: A mediator-based multi-agent system for cooperative information gathering from the world wide web. In *Proceedings of the Fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000)*, pages 331–334, Manchester, UK, April 2000. The Practical Application Company Ltd.
- [8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, USA, 1988.
- [9] M. van Lent, J. E. Laird, J. Buckman, J. Hartford, H. S., K. Steinkraus, and R. Tedrake. Intelligent agents in computer games. In *Proceedings of the National Conference on Artificial Intelligence*, pages 929 – 930, Orlando, FL, July 1999. AAAI.
- [10] M. Wellman. A general-equilibrium approach to distributed transportation planning. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 282–289, Cambridge MA, 1992. The AAAI Press.