

User Type Identification in Virtual Worlds

Ruck Thawonmas, Ji-Young Ho, and Yoshitaka Matsumoto

Introduction

In this chapter, we discuss an approach for identification of user types in virtual worlds. A popular form of the virtual world is a massively multiplayer online game (MMOG). MMOGs provide fast-growing online communities [1], and managing a large-scale virtual community implies many challenges, such as identification of user types, social structures, and virtual economic mechanisms [2]. In this chapter, we address the challenge on identification of user types. It is very important to grasp users' needs and to satisfy them through furnishing appropriate contents for each user or each specific group of users.

In virtual worlds, four user types are typically identified by their characteristics, namely, "killer," "achiever," "explorer," and "socializer" [3]. Killer-type users just want to kill other users and monsters with the tools provided. Achiever-type users set their main goal to gather points or to raise levels while the explorer-type user want to find out interesting things about the virtual world and then to expose them. Socializer-type users are interested in relationships among users. Following this categorization, a typical use of user-type identification results can be depicted as in Fig. 1. In this figure, users are categorized into predefined types based on appropriate selected features from the logs, and are provided contents according to their favorites. Thereby, the users should enjoy the virtual world more and hence stay longer. As a first step toward use of real virtual world data, we demonstrate our approach using a PC cluster-based MMOG simulator.

The work presented in this chapter is divided into two phases, namely, modeling and identification. In the modeling phase, many types of user agents with different characteristics are modeled using the above MMOG simulator. By user agents, we mean agents that imitate user characters in real MMOGs. The user

Intelligent Computer Entertainment Laboratory, Department of Human and Computational Science, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan. e-mail: ruck@ci.ritsumeiji.ac.jp

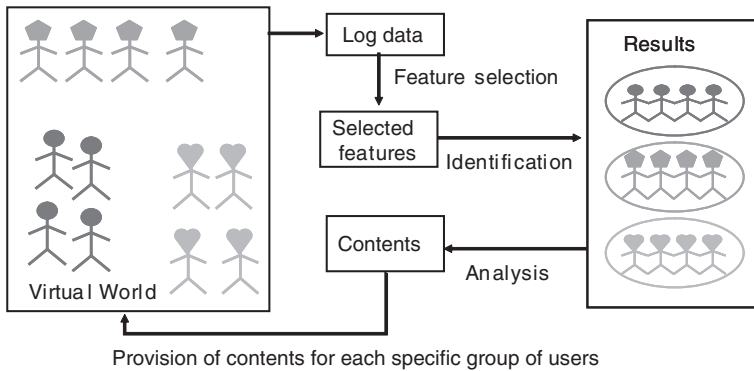


FIG. 1. Typical use of user-type identification results

agents reside in and migrate among multiple worlds, each world running on a PC node. A world also accommodates monsters, representing nonplayer characters in real MMOGs, that can kill (or be killed by) user agents.

In the identification phase, the task is to correctly identify the type of a given user agent from its log. To perform this task, two technical issues are discussed. The first one is feature selection, namely, selection of input features from log data. The other one is classifier selection, namely, selection of a classifier for identifying a given user agent to a particular type based on the selected input features.

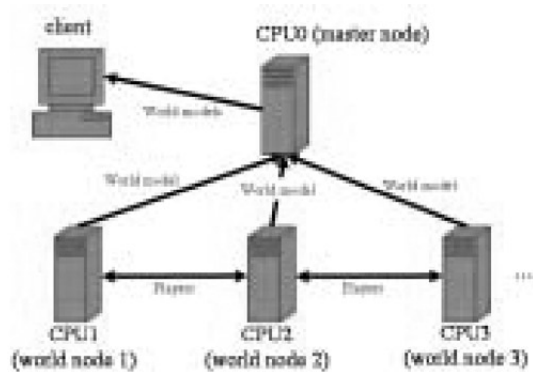
MMOG Simulator and Agent Modeling

The PC cluster-based MMOG simulator that we use is Zereal [4]. Zereal is a multiagent simulation system [5]. It can simulate multiple worlds simultaneously, running each world on a different PC node.

Figure 2 shows the architecture of Zereal. It is composed of one master node and multiple world nodes. The master node collects the current status (world model) of each world and forwards this information to a client computer for visualization or data analysis. A world node simulates all objects such as user agents and monster agents. Other objects include food items and potion items for recovering stamina, and key items for opening a door in order to leave the current world.

In the version of Zereal that we licensed from the Zereal developing team, three types of user agents, namely, Killer, Markov Killer, and Plan Agent, are provided. Each type has six common actions, namely, Walk, Attack, PickFood, PickPotion, PickKey, and LeaveWorld, but each type is designed to have different behavior described as follows:

FIG. 2. Zereal architecture



- Killer puts the highest priority on killing monsters.
- Markov Killer gets as many items as possible to be stronger. User agents of this type also kill monsters, but attack monsters according to the corresponding state-transitional probability.
- Plan Agent finds a key and leaves the current world.

Killer, Markov Killer, and Plan Agent correspond to, to some extent, “killer,” “achiever,” and “explorer,” respectively, as described earlier.

To observe activities in the artificial societies, visualization tools are crucial for ■■ (MASs). We have developed such a tool called ZerealViewer. Although not yet fully functional, a screen shot of the ZerealViewer when one world is simulated is shown in Fig. 3.

Figure 4 shows a typical virtual world log sent to the client from the master node for data analysis. The first and the second columns in the log indicate the simulation time steps and the real clock time, respectively. The third column shows the agent identifier numbers with the most upper digit(s) indexing the current world node. The fourth column represents agent actions, and the fifth and sixth columns show the coordinates in the world before and after such actions, respectively. The last column gives information on the types of agents.

User Identification

User identification of a given user agent is performed merely from its log. In our case, although type information is already available in the log, this information is not used.

Feature Selection

Two types of sequences, action sequences and item sequences, are generated by different algorithms. Action sequences [6] are generated from log data by extraction of action information. Items sequences [7] are generated by the following algorithm:



FIG. 3. Screen shot of ZerealViewer

```

20|2003-5-28|12:12:24|1000008|Walk|(18,7)|(17,6)|MarkovKiller
20|2003-5-28|12:12:24|1000009|PickFood|(18,12)|(19,12)|MarkovKiller
20|2003-5-28|12:12:24|1000010|Walk|(24,8)|(23,9)|MarkovKiller
20|2003-5-28|12:12:24|1000011|Walk|(16,5)|(16,4)|MarkovKiller
20|2003-5-28|12:12:24|1000019|Removed|(29,10)|.|Killer
20|2003-5-28|12:12:24|1000013|Attack|(30,10)|(29,10)|Monster
20|2003-5-28|12:12:24|1000016|Attack|(39,9)|(39,10)|Monster
20|2003-5-28|12:12:24|1000018|Walk|(27,10)|(28,10)|PlanAgent
21|2003-5-28|12:12:24|1000007|Walk|(32,11)|(31,12)|MarkovKiller
21|2003-5-28|12:12:24|1000008|Walk|(17,6)|(16,5)|MarkovKiller
21|2003-5-28|12:12:24|1000009|Walk|(19,12)|(18,11)|MarkovKiller
21|2003-5-28|12:12:24|1000010|Walk|(23,9)|(23,10)|MarkovKiller
21|2003-5-28|12:12:24|1000013|Walk|(30,10)|(29,10)|Monster
21|2003-5-28|12:12:24|1000014|Walk|(31,10)|(30,10)|Monster
21|2003-5-28|12:12:24|1000016|Attack|(39,9)|(39,10)|Monster
21|2003-5-28|12:12:24|1000026|Attack|(39,10)|(39,9)|Killer

```

FIG. 4. Typical virtual world log

- For monster items, if a user agent attacks a particular monster, add one monster item to the item sequence of that user agent. If the user agent attacks the same monster many times, only one monster item is added.
- For food, potion, and key items, if a user agent picks food, potion, or key, add one food, potion, or key item to the item sequence of that user agent, respectively.
- For door items, if a user agent leaves the world through a door, add one door item to the item sequence of that user agent.

Figures 5 and 6 show the resulting action sequences and item sequences, respectively. In addition, tables 1 and 2 show the relative frequencies of user agent actions and user agent items, respectively. Because the tendencies of agent behaviors can be seen from the frequencies of action sequences and item sequences, it is possible to identify user agents based on this kind of information.

We apply the following algorithm to action sequences to generate the input features for a classifier discussed in the next section.

TABLE 1. Relative frequencies (columnwise) of user agent actions

PC types	Walk	Attack	PickFood	PickPotion	PickKey	LeaveWorld
Killer	L	H	M	M	L	L
Markov Killer	M	M	H	H	M	M
Plan Agent	H	L	L	L	H	H

L, Low; M, medium; H, high

TABLE 2. Relative frequencies (columnwise) of user agent items

PC types	Monster	Food	Potion	Key	Door
Killer	H	M	M	L	L
Markov Killer	M	H	H	M	M
Plan Agent	L	L	L	H	H

- Step I: For each user agent, sum the total number of each action that the user agent performed.
- Step II: For each user agent, divide the result of each action in Step I by the total number of actions that the user agent performed.
- Step III: For each user agent, divide the result of each action in Step II by that of the agent who most frequently performed the action.

The feature-selection algorithm for the item sequences is the same as the one above, except that action is replaced by item, and performed is replaced by acquired. Tables 3, 4, and 5 show typical results of steps I, II, and III, respectively, for both action features and item features.

Classifier Selection

Here we adopt adaptive memory-based reasoning (AMBR) as the classifier in our experiments. AMBR [8] is a variant of memory-based reasoning (MBR). Given an unknown data to classify, MBR [9] performs majority voting of the labels (user types in our case) among the k nearest neighbors in the training data set, where the parameter k has to be decided by the user. On the contrary, AMBR is MBR with k initially set to 1; when ties in the voting occur, it increments k accordingly until ties are broken. Figure 7 depicts the concept of AMBR with three types of data represented by circles, triangles, and squares. To predict the type of unknown data represented by the cross, the procedure attempts to find the nearest neighbor (Fig. 7a), but a tie occurs with two circles and two squares. According to the procedure, after neglecting the triangle type that is not in the tie, k is increased to 5 (Fig. 7b) by which five circles and three squares are found in the next step. Finally the unknown data is predicted as a circle.

TABLE 3. Typical results of step I: total number of actions and items

PC types	Action features					Item features					
	Walk	Attack	PickFood	PickPotion	PickKey	LeaveWorld	Monster	Food	Potion	Key	Door
Killer 1	67	92	2	0	0	0	8	2	0	0	0
Killer 2	93	104	0	1	2	0	11	0	1	2	0
Markov Killer 1	107	1	6	2	0	0	1	6	2	0	0
Markov Killer 2	177	11	4	7	1	0	1	4	7	1	0
Plan Agent 1	113	0	0	0	8	1	0	0	0	8	1
Plan Agent 2	119	0	1	0	4	0	0	1	0	4	0

TABLE 4. Typical results of step II: action and item frequencies for each agent

PC types	Action features					Item features					
	Walk	Attack	PickFood	PickPotion	PickKey	LeaveWorld	Monster	Food	Potion	Key	Door
Killer 1	0.4161	0.5714	0.0124	0	0	0	0.8000	0.2000	0	0	0
Killer 2	0.4650	0.5200	0	0.0050	0.0100	0	0.7857	0	0.0714	0.1429	0
Markov Killer 1	0.9224	0.0086	0.0517	0.0172	0	0	0.1111	0.6667	0.2222	0	0
Markov Killer 2	0.8850	0.0550	0.0200	0.0350	0.0050	0	0.0769	0.3077	0.5385	0.0769	0
Plan Agent 1	0.9262	0	0	0	0.0656	0.0082	0	0	0	0.8889	0.1111
Plan Agent 2	0.9597	0	0.0081	0	0.0323	0	0	0.2000	0	0.8000	0

TABLE 5. Typical results of step III: action and item frequencies among all agents

PC types	Action features					Item features					
	Walk	Attack	PickFood	PickPotion	PickKey	LeaveWorld	Monster	Food	Potion	Key	Door
Killer 1	0.4336	1.0000	0.2402	0	0	0	1.0000	0.3000	0	0	0
Killer 2	0.4845	0.9100	0	0.1429	0.1525	0	0.9821	0	0.1327	0.1607	0
Markov Killer 1	0.9612	0.0151	1.0000	0.4926	0	0	0.1389	1.0000	0.4127	0	0
Markov Killer 2	0.9222	0.0963	0.3867	1.0000	0.0762	0	0.0962	0.4615	1.0000	0.0865	0
Plan Agent 1	0.9651	0	0	0	1.0000	1.0000	0	0	0	1.0000	1.0000
Plan Agent 2	1.0000	0	0.1559	0	0.4919	0	0	0.3000	0	0.9000	0

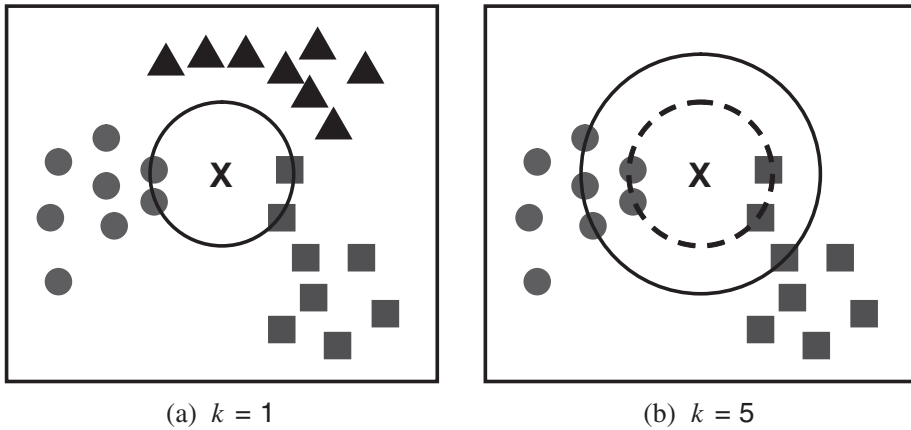


FIG. 7. Concept of Adaptive Memory Based Reasoning

Experiments

Any classifier should be able to correctly identify unknown data not seen in the training data. This ability is called generalization ability. To approximate the generalization ability, we use the leave-one-out method [10]. In the leave-one-out method, supposing that the total number of available data is M , first, data number 1 is used for testing and the other data are used for training the classifier of interest. Next, data number 2 is used for testing and the other data are used for training the classifier. The process is iterated a total of M times. In the end, the averaged recognition rate for test data is computed, and is used to indicate the generalization ability of the classifier.

For experiments, log data were generated by running ten independent Zereal games with 500 simulation-time steps. In each game, we simulated 100 user agents of each type, 100 monsters, and 100 items for each of the other objects. For the generated log data, we conducted the feature selection algorithms discussed in the previous section, and obtained the input features to AMBR for each sequence type.

Figure 8 shows the recognition rates, indicating the generalization ability, for each type of input feature over ten Zereal games. Based on these results, we performed hypothesis test (t -test) for the difference in the recognition rates with 99% confidence. The resulting t value and P value are -4.54 and 0.07% , respectively. As a result, the difference in the recognition rates is statistically significant, and the item-based features outperform the action-based features in terms of generalization ability.

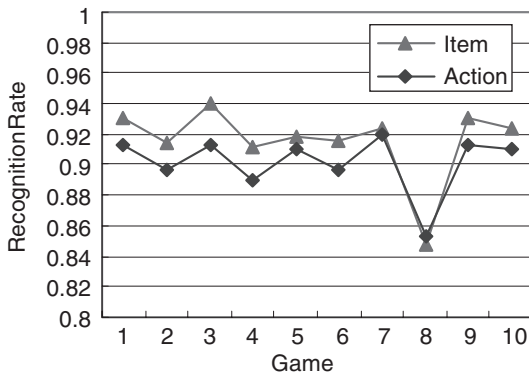


FIG. 8. Recognition rates for each type of input feature

Conclusions

In this chapter we have presented an effective approach for identification of user types in virtual worlds. Two types of input features were discussed, action-based features and item-based features. The former type uses the information on the frequency of each type of action that each user performed. The latter one uses the information on the frequency of each type of item that each user acquired. AMBR, adopted as the classifier, could successfully identify the type of unknown user agents. In addition, it could give higher performance with the item-based features. In future work, we plan to conduct experiments using agents with more complicated behaviors and to investigate use of order information in either action sequences or item sequences. Eventually, we will apply our findings to real virtual world data.

Acknowledgments. Ruck Thawonmas was supported in part by the Ritsumeikan University's Kyoto Art and Entertainment Innovation Research, a project of the 21st Century Center of Excellence Program funded by the Japan Society for Promotion of Science. Ji-Young Ho was supported by a scholarship from the Ministry of Education, Culture, Sports, Sciences, and Technology, Japan.

References

1. Jarett A, Estanislao J, Dunin E, et al. (2003) IGDA Online Games White Paper, 2nd ed. ■■
2. Thawonmas R, Yagome T (2004) Application of the artificial society approach to multiplayer online games: a case study on effects of a robot rental mechanism. Proceedings of the 3rd International Conference on Application and Development of Computer Games (ADCOG 2004), Hong Kong
3. Bartle R (1996) Hearts, clubs, diamonds, spades: players who suit MUDs. The Journal of Virtual Environments 1:■■■-■■■

4. Tveit A, Rein O, Jorgen VI, et al. (2003) Scalable agent-based simulation of players in massively multiplayer online games. Proceedings of the 8th Scandinavian Conference on Artificial Intelligence (SCAI 2003), Bergen, Norway
5. Epstein J, Axtell R (1996) Growing artificial societies: social science from the bottom up. MIT, Brookings, MA
6. Thawonmas R, Ho JY, Matsumoto Y (2003) Identification of player types in massively multiplayer online games. Proceedings of the 34th Annual Conference of International Simulation and Gaming Association (ISAGA 2003), Chiba, Japan, pp 893–900
7. Ho JY, Matsumoto Y, Thawonmas R (2003) MMOG player identification: a step toward CRM of MMOGs. Proceedings of the 6th Pacific Rim International Workshop on Multi-Agents (PRIMA2003), Seoul, Korea, pp 81–92
8. Ho JY, Thawonmas R (2004) Episode detection with vector space model in agent behavior sequences of MMOGs. Proceedings of Future Business Technology Conference 2004 (FUBUTEC'2004), Fontainebleau, France, pp 47–54
9. Berry M, Linofi G (1997) Data mining techniques—for marketing, sales, and customer support. Wiley, New York
10. Weiss S, Kulikowski C (1991) Computer systems that learn. Morgan Kaufmann, San Mateo, CA