Proc. of IFIP 4th International Conference on Entertainment Computing (ICEC 2005), Sep. 2005, Sanda, Japan, published in Lecture Notes in Computer Science, Fumio Kishino et al. (Eds.), vol. 3711, pp. 126-135.

Aggregation of Action Symbol Sub-sequences for Discovery of Online-Game Player Characteristics Using KeyGraph

Ruck Thawonmas* and Katsuyoshi Hata

Intelligent Computer Entertainment Laboratory,
Department of Human and Computer Intelligence,
Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
ruck@ci.ritsumei.ac.jp

Abstract. Keygraph is a visualization tool for discovery of relations among text-based data. This paper discusses a new application of KeyGraph for discovery of player characteristics in Massively Multiplayer Online Games (MMOGs). To achieve high visualization ability for this application, we propose a preprocessing method that aggregates action symbol sub-sequences of players into more informative forms. To verify whether this aim is achieved, we conduct an experiment where human subjects are asked to classify types of players in a simulated MMOG with KeyGraphs using and not using the proposed preprocessing method. Experimental results confirm the effectiveness of the proposed method.

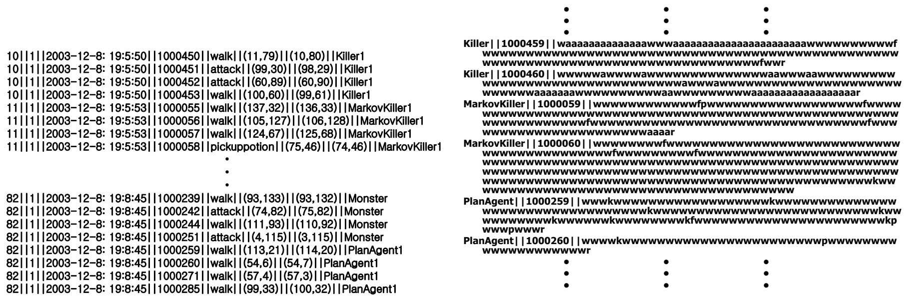
1 Introduction

The market size of Massively Multiplayer Online Games (MMOGs) continues to experience surging growth. According to the Themis Group [1], estimated worldwide revenues of MMOGs will rise from 1.30 Billion USD in 2004 to 4.10 Billion USD in 2008, and to 9 Billion USD in 2014. At the same time, competitions among MMOGs are also becoming very high. Besides acquisition of new players, retention of current players is also very important. For player retention, tools are needed that discover player characteristics, so that tailored contents or supports can be provided to players. One of such tools is KeyGraph [2] that was originally proposed for extracting keyword terms in a document. Its underlying concept is based on a building construction metaphor. KeyGraph has been later applied to visualizing the relations among Web pages, among products in markets, and among earthquake faults, etc. [3].

In this paper, we apply KeyGraph to discovery of MMOG player characteristics. However, typical MMOG player data consist mainly of sequences of

* The author has been supported in part by Ritsumeikan University's **Kyoto Art and Entertainment Innovation Research**, a project of the 21^{st} Century Center of Excellence Program funded by Ministry of Education, Culture, Sports, Science and Technology, Japan; and by Grant-in-Aid for Scientific Research (C), Number 16500091, the Japan Society for Promotion of Science.

F. Kishino et al. (Eds.): ICEC 2005, LNCS 3711, pp. 126–135, 2005.



(a) (b)

Fig. 1. (a) MMOG player data from an MMOG simulator Zereal and (b) action symbol sequences of players

action symbols (Fig. 1) that are not fit for KeyGraph. This is because an action symbol in such sequences does not represent important concepts, but a group of them might do, in analogy to characters and terms in a document, respectively. We therefore propose a preprocessing method for iteratively aggregating sub-sequences of consecutive action symbols into more informative forms. Experimental results from simulated MMOG player data show that KeyGraph using preprocessed MMOG player data has a higher visualization ability to distinguish player types than KeyGraph using the original player data.

2 KeyGraph

Here rather than giving a detailed explanation of it, we briefly describe an outline of KeyGraph. KeyGraph consists of three major components derived based on building construction metaphor. Each component is described as follows:

Foundations – sub-graphs of highly associated and frequent terms that represent basic concepts in the data,
Roofs – terms that are highly associated with foundations,
Columns – associations between foundations and roofs that are used for extracting keywords, i.e., main concepts in the data.

In KeyGraph, associations between terms are the co-occurrence among them in same sentences, and keywords are the terms in either foundations or roofs that are connected to strong columns. In addition, foundations are depicted by solid lines and their touching black nodes, columns by dotted lines, roofs excluding those in the foundations by red nodes, and keywords by double circles.

Fig. 2 shows an example of KeyGraph[1] when it is applied to the text data taken from Section 1 of this paper. From the result, one can see that there are two main concepts in Section 1, i.e., the concept about "KeyGraph for MMOG

[1] In this paper, we use a commercial KeyGraph tool developed by Kozo Keikaku Engineering Inc.

player data" from 5 keywords on the foundation, and the other about "estimated billion USD revenues of MMOGs" from the cluster of 3 keywords and a number of roofs.

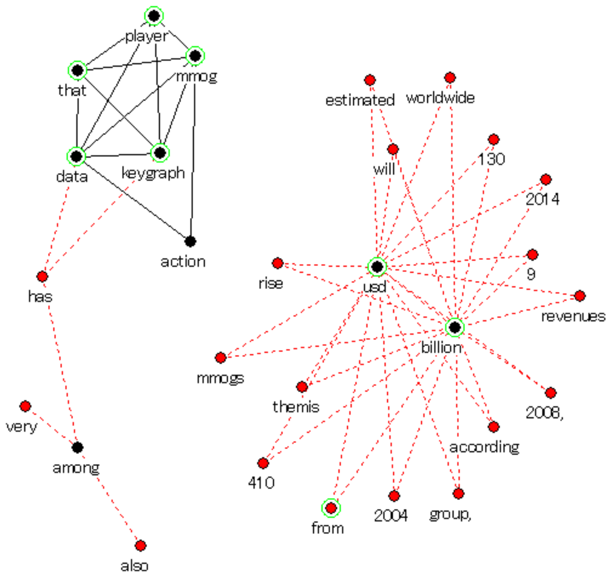


Fig. 2. KeyGraph applied to Section 1 of this paper

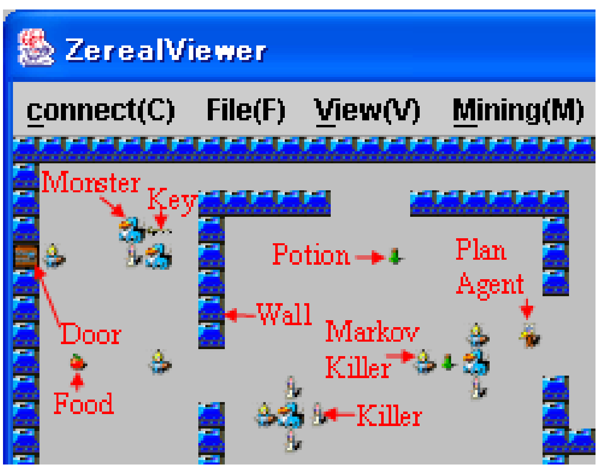


Fig. 3. Screenshot of a game world in Zereal

3 MMOG Simulator and Player Modeling

To obtain MMOG player data, we use Zereal [4], a Python-based multiple agent simulation system. Figure 3 shows the screen shot of a game world in Zereal. Zereal was designed for testing player modeling, artificial intelligence, and data analysis techniques for MMOGs. We focus on three types of player agents, as done in [5], i.e., Killer, InexperiencedMarkovKiller, and Experienced-MarkovKiller, each having 9 actions listed in Table 1. Three player types have different characteristics and different intelligence levels, as summarized in the following descriptions; the last two use a probability matrix following a Markov model for deciding the next action:

- **Killer (K)** roams among game worlds and pursues the closest player or monster and kill them; it has no sociability and thus does not chat with others.
- **InexperiencedMarkovKiller (IMK)** randomly equally attempts all possible actions in a given situation; it models a novice player.
- **ExperiencedMarkovKiller (EMK)** prefers particular actions over others in a given situation and tends to attack monsters nearby; this player type models a veteran player.

In this study, we ran 16 Zereal game worlds, 300 simulation-time cycles each. In each game world, there were 50 player agents for each type, 50 monster agents, and 50 items for each game object (food, potion, and key).

Figure 4 shows KeyGraphs for the three player types when action symbol sequences of players from Zereal are not preprocessed with our method, discussed in the next section. Though their foundations are slightly different, these KeyGraphs have the same keywords, except the KeyGraph for **K** that has no keywords associated with "talk". As a result, it is hard to distinguish one player type from others with these KeyGraphs.

Table 1. Summary of player agent actions in Zereal

Action	Symbol	Description	Precondition
walk	w	walks from one place to another	not blocked by other objects
attack	a	attacks other players or monsters	targets available nearby
talk	t	talks to other players	other players available nearby
pick up food	f	picks up food items	food items available nearby
pick up potion	p	picks up potion items	potion items available nearby
pick up key	k	picks up key items	key items available nearby
leave the world	l	leaves the current world through a door	has a key of the door
enter the world	e	enters the current world through a door	has a key of the door
removed	r	removed from the game	hit points reaches 0

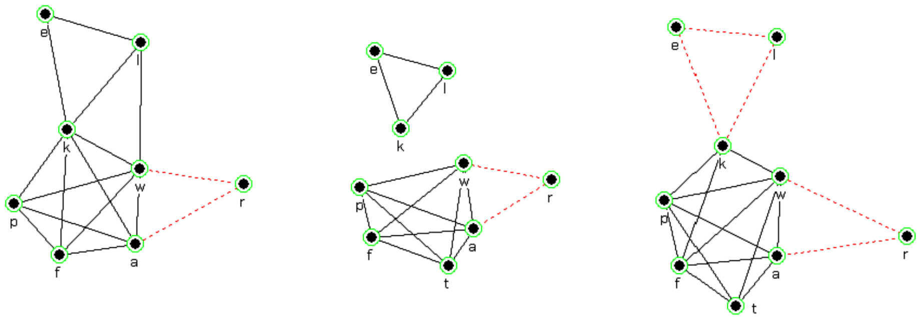


Fig. 4. KeyGraphs for the three player types, from left to right, (a) Killer, (b) InexperiencedMarkovKiller, and (c) ExperiencedMarkovKiller, when player data are not preprocessed with the proposed method

4 Action Symbol Sub-sequence Aggregation

As shown in Fig. 1, the action symbol sequence of a player contains many repetitive portions. One action symbol does not represent that much information on agent characteristics. On the contrary, we conjecture that a sub-sequence of consecutive action symbols, for example, "..wwaa..aaww.." as frequently seen in **K** indicates that the objective of this type of player is to look for targets to attack. Based on this conjecture, we propose an algorithm that aggregates a frequent sub-sequence of consecutive symbols into a more informative symbol called a chained symbol. The proposed algorithm interactively performs aggregation of frequent sub-sequences until they are no longer found.

The proposed algorithm below mainly consists of two parts, i.e., frequent sub-sequence detection (Step 1-2) and frequent sub-sequence aggregation (Step 3-4). The rest of the algorithm (Step 5-6) is for formatting final results.

Step 1. Obtain n-grams[2] from each sequence, and calculate the occurrence frequency in all sequences of each n-gram.

Step 2. Flag all n-grams whose occurrence frequency is T times above the average.

Step 3. For each sequence, find from left to right any sub-sequence that matches with one of the flagged n-grams, and aggregate that sub-sequence into a new symbol called a chained symbol.

Step 4. Count a chained symbol as one symbol in n-grams and repeat Step 1 to Step 3 until there are no n-grams to flag.

Step 5. For each chained symbol in a sequence, replace a portion having consecutive identical action symbols by the capital letter of that action symbol.

[2] An n-gram is an n-long sub-sequence of consecutive symbols. It is called a bi-gram if $n = 2$.

Step 6. For each sequence, put spaces between chained symbols and remaining action symbols.

The example below shows the result from the above algorithm with $n = 2$ and $T = 2$, for the action symbol sequences:

```
sequence #1 -- wwwwwwwwwwawwpwwa
sequence #2 -- wawwwwwwwwfwpwwa
sequence #3 -- wwaawwawwwwwwaww
```

At Step 1, the following bi-grams and their occurrence frequencies are obtained.

```
ww:27, wa:7, aw:5, wp:2, pw:2, wf:1, fw:1, aa:1
```

At Step 2, since the average occurrence frequency is 5.75, only the bi-gram ww, the occurrence frequency of which is above 2 times of the average, is flagged.

At Step 3, the resulting sequences are as follows:

```
sequence #1 -- [ww][ww][ww][ww][ww]a[ww]p[ww]a
sequence #2 -- wa[ww][ww][ww][ww]fwp[ww]a
sequence #3 -- [ww]aa[ww]a[ww][ww][ww]a[ww]
```

where and henceforth a chained symbol is surrounded by brackets for the sake of illustration.

Step 1 at the next iteration gives the following bi-grams and their occurrence frequencies.

```
[ww][ww]:9, [ww]a:6, a[ww]:5, p[ww]:2, [ww]p:1,
wa:1, [ww]f:1, fw:1, wp:1, aa:1
```

At Step 2, since the average occurrence frequency is 2.8, the two bi-grams [ww][ww] and [ww]a, the occurrence frequency of which is above 2 times of the average, are flagged.

At Step 3, the resulting sequences are as follows:

```
sequence #1 -- [wwww][wwww][wwa][ww]p[wwa]
sequence #2 -- wa[wwww][wwww]fwp[wwa]
sequence #3 -- [wwa]a[wwa][wwww][wwa][ww]
```

Since there are no bi-grams for being flagged at the next iteration, the algorithm goes to Step 5.

At Step 5, the resulting sequences are as follows.

```
sequence #1 -- [W][W][Wa][W]p[Wa]
sequence #2 -- wa[W][W]fwp[Wa]
sequence #3 -- [Wa]a[Wa][W][Wa][W]
```

At Step 6, the following sequences are obtained.

```
sequence #1 -- [W] [W] [Wa] [W] p [Wa]
sequence #2 -- w a [W] [W] f w p [Wa]
sequence #3 -- [Wa] a [Wa] [W] [Wa] [W]
```

5 Experimental Results

We set n and T to 2 and 10, respectively, for the proposed algorithm.

5.1 KeyGraphs for Player Data Preprocessed with the Proposed Method

Figures 5(a), 5(b), and 5(c) show KeyGraphs for **K**, **IMK**, and **EMK**, respectively, when action symbol sequences of players are preprocessed with the proposed method. Each KeyGraph represents well the characteristics of the corresponding player type. They are summarized as follows:

Fig. 5(a): It is the only KeyGraph that has keywords associated with "attack" on the foundation. In addition, it has no nodes associated with "talk". Apart from the foundation, there is a cluster of keywords associated with "leave the world", "enter the world", and "pick up key".

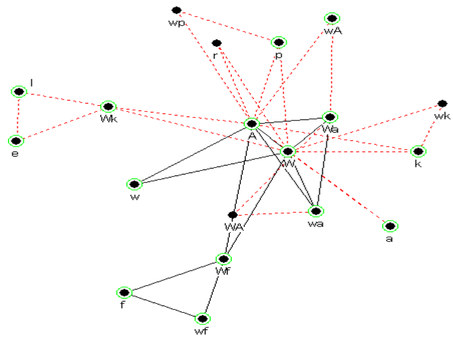
Fig. 5(b): Its foundation is formed by keywords associated with "walk" and "talk". Apart from the foundation, there is a cluster of keywords associated with "attack" and "walk" in the right top corner. Other keywords outside of the foundation do not form clusters among themselves, showing randomness in selection of actions.

Fig. 5(c): Compared with the above two KeyGraphs, there are more clusters of keywords here. This implies that they are patterns in selection of actions. Such clusters are a cluster associated with "pick up potion" and "walk" in the left bottom corner, a cluster associated with "pick up food" and "walk" in the left middle part, and a cluster associated with "attack" and "walk" in the right middle part.

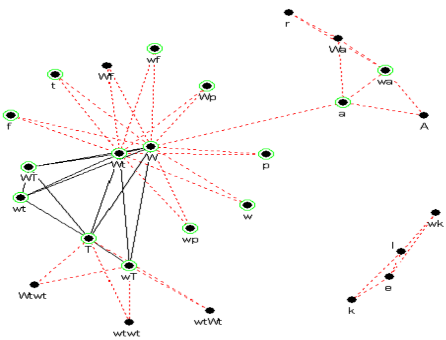
5.2 Player-Type Classification by Human Subjects

Since KeyGraph is a visualization tool, we conducted an experiment to see whether human subjects can classify resulting KeyGraphs correctly according to player types when the proposed preprocessing method was and was not used. First, we had 10 subject read Section 2 and Section 3, with the last paragraph (on Fig. 4) being excluded, of this paper. This was done in order to provide them the necessary information on KeyGraph and player types. Then, each subject was shown in a random order the group of KeyGraphs in Fig. 4 and the group of KeyGraphs in Fig. 5. For each group, each KeyGraph was also shown in a random order to the subject. Finally, they were asked to label the player type of each KeyGraph in a given group.

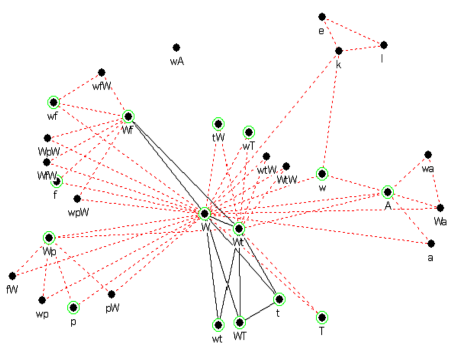
Tables 2 and 3 show the classification results not using and using the proposed method, respectively. The classification rate of the latter is higher than that of the former for **IMK** and **EMK**. For **K**, they have the same classification rate, as expected since most subjects should easily detect non-presence of nodes associated with "talk".



(a)



(b)



(c)

Fig. 5. KeyGraphs for the three player types, from top to bottom, (a) Killer, (b) InexperiencedMarkovKiller, and (c) ExperiencedMarkovKiller, when player data are preprocessed with the proposed method

Table 2. Player-type classification results by human subjects for KeyGraphs in Fig. 4

Answers\Types	**K**	**IMK**	**EMK**
K	9	0	0
IMK	0	3	3
EMK	0	3	3
Not Sure	1	4	4
Classification Rate	0.9	0.3	0.3

Table 3. Player-type classification results by human subjects for KeyGraphs in Fig. 5

Answers\Types	**K**	**IMK**	**EMK**
K	9	1	0
IMK	0	7	3
EMK	1	2	7
Not Sure	0	0	0
Classification Rate	0.9	0.7	0.7

6 Conclusions

We have shown in this paper that KeyGraph is a powerful tool for discovery of the player characteristics of **K**, **IMK**, **EMK** in Zereal, provided that player data are preprocessed with the proposed method. Example applications for real MMOGs include discovery of the characteristics of players who belong to a same social group (guild), who have a same level, who get bored soon after starting the game, etc.; once such characteristics are discovered, better communication tools or proper levels of game difficulties can be provided to corresponding players. After incorporation of DNA analyzing techniques, we plan to test our approach with real data from an edutainment online game called The ICE [6], under development at the authors' laboratory, to be released soon to students of our university.

References

1. Alexander, K., Bartle, R., Castronova, E., Costikyan, G., Hayter, J., Kurz, T., Manachi, D., Smith, J.: The Themis Report on Online Gaming 2004. www.themis-group.com/reports.phtml (2004)
2. Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: automatic indexing by co-occurrence graph based on building construction metaphor. Proc. Advanced Digital Library Conference (IEEE ADL'98) (1998) 12-18
3. Ohsawa, Y., McBurney, P. (eds.): Chance Discovery - Foundation and Its Applications. Springer Verlag (2003)
4. Tveit, A., Rein, O., Jorgen, V.I., Matskin, M.: Scalable Agent-Based Simulation of Players in Massively Multiplayer Online Games. Proc. the 8th Scandinavian Conference on Artificial Intelligence (SCAI2003), Bergen, Norway, (2003)

5. Matsumoto, Y., Thawonmas, R.: MMOG Player Classification Using Hidden Markov Models. In: Rauterberg, M. (ed.): Lecture Notes in Computer Science, vol. 3166, 429-434 (Third International Conference on Entertainment Computing (ICEC 2004), Eindhoven, The Netherlands, Sep. 2004)
6. Thawonmas, R., Yagome, T.: Application of the Artificial Society Approach to Multiplayer Online Games: A Case Study on Effects of a Robot Rental Mechanism. Proc. of the 3rd International Conference on Application and Development of Computer Games (ADCOG 2004), HKSAR, 12-17 (2004)