A NEW CONCEPT IN THE REAL TIME WEB GAMES

How to develop the highly real-time Web games

Yoshihiro KAWANO, Masahiro MIYATA
Intec NetCore, Inc.,Tokyo, Japan
kawano@inetcore.com
Department of Computer and Information Sceince, Ibaraki University, Ibaraki, Japan
t44061y@hcs.ibaraki.ac.jp

Dai HANAWA, Tatsuhiro YONEKURA
Department of Computer and Information Science, Faculty of Science and Technology, Seikei University,Tokyo, Japan
hanawa@st.seikei.jp
Department of Computer and Information Sceince, Ibaraki University, Ibaraki, Japan
yone@mx.ibaraki.ac.jp

Keywords: Web game, Real-time, Ajax, Information Extraction Layer, Dead Reckoning, Allocated Topographical Zone

Abstract: The online game has been rapidly popularized recently, with the improvement of the network speed and the computer performance. Accordingly, the problem of network latency has been significant in applications with a requirement of high interactivity such as the action game and real-time sports game. In general, a server-client type of architecture may be employed in such games, namely a Distributed Virtual Environment (DVE). In this paper, we focus on the Web system that is widely widespread as the infrastructure of this type. However, strategy of designing web games is not explicitly established. We propose a new concept to design the real-time Web game. In the former paper, we introduced new schemes, namely, Dead Reckoning and Allocated Topographical Zone. These are what we employed in the real-time Web games to overcome limitation of HTTP communication. As an example of the application, Web based real-time avatar operation game is implemented. As a result, we confirmed that our concept may sufficiently expand to the various types of the real-time Web applications.

1 INTRODUCTION

Recently, the online game has been rapidly popularized, with the improvement of the network speed and the computer performance. Accordingly, the problem of network latency has been significant in applications with a requirement of high interactivity such as the action game and real-time sports game (Takemura, 1999). In general, a server-client type of architecture may be employed in such games, namely a Distributed Virtual Environment (DVE hereafter) (Takemura, 1999).

As the example of the highly real-time online game, there are Counter-Strike (Valve Corporation, 2007) that is FPS (First Person Shooter) type and Age of Empires III (Microsoft Corporation, 2007) that is RTS (Real-time Strategy) type. And, though it is only for the particular ISP (Internet Service Provider) user, SHIN-SANGOKU MUSOU BB (KOEI Co., Ltd., 2007) achieves the highly real-time multiplayer action game. However, in order to play these kinds of online game, the user has to install special client software which is dedicated only for that game. Moreover, if the software uses the particular port-number, the data may not be transmitted over the firewall.

In this paper, we focus on the Web that is widely widespread. When a game is actualized on the Web, the user can play it only with a browser, and then it will lead the wide use and the increment of the number of the users. On the other hand, by using HTTP to transmit the data, the transmission will not be limited by the firewall. And moreover, this scheme will be able to be applied not only for the game but also efor ducation to provide some experiences. Here, there is a Web game named Dinoparc (Motion-Twin, 2005), that is considered as an example of the multiplayer online game on the

Web though highly real-time Web game is not reported.

On the other hand, JAVA based MMORPG (Massively Multi-player Online Role Playing Game) on the Web called RuneScape is available (Jagex Ltd., 2007) though the game seems not to allow requirement of highly real-time interaction. And, Shockwave based MMOFPS type of real-time Web game called Tank Ball2 is available (Maid Marian Entertainment, 2004). However, these games are doubtful about consistency of collision detection between tank and ball, in spite of that the collision detection is the most critical factor. Therefore, in the real-time game, shared objects cannot be controlled jointly by all the users because of the temporal difference between the local and the remote avatars (virtual player) in terms of their status in reference to each other (Hashimoto, Sheridan and Noyes, 1986).

Thus, this paper studies the way to improve consistency for all states, including those of the avatars and shared objects in a real-time Web game in which the shared field is equally accessible to and controllable by each terminal, without sacrificing throughput. Here, we adopted Virtual Ball Game (VBG hereafter) as an example of real-time Web game. To construct a VBG, in which the objects shared by the players have the physical attributes of location, orientation and velocity, a significant issue is that of how the ownership (decision of which terminal owns the key to control an object) and the attributes for all avatars and objects should be kept consistent among terminals. In this paper, we propose a new concept to design the real-time Web game.

2 REQUIREMENTS OF THE REAL TIME WEB GAMES

There are three requirements to achieve VBG type of the real-time Web games.

(1) Consistency for the attributes of avatars
(2) Throughput to the operation of users
(3) Speed and Accuracy of collision detection about the shared object

Here, the data transmission by using HTTP because of Web has to satisfy three requirements in the above. However, HTTP has the critical issue for real-time game that is transmission lag, limitation of bandwidth, and jitter. Then, our motivation is to overcome the critical issue above, and to achieve real-time game on the Web.

From the requirements of (1) and (2), game logic part should separate from data transmission part. Then, we can consider about the use of Ajax (Asynchronous JavaScript + XML) (OpenAjax Alliance, 2007). Ajax has advantages as follows.

- Data can be transmitted asynchronously.
- Game logic can be executed on the local client side.
- Only a necessary part can be changed.

Written in above, Ajax is necessary to achieve the real-time Web game. By using Ajax, each client executes MVC (Modeling-View-Control) loop and organizes a virtual world by oneself, and then the load of the server can be decentralized. Therefore, in this paper, we set the principal object to the Web game that uses Ajax. And then, in our concept, the information represented on each terminal may have inconsistency, because the network is considered as a virtual P2P model. Then, synchronization between terminals is not easily available because such a network has no server for synchronization management, as does a server-client network. Therefore, it is necessary to keep consistency of the presented information by using Dead Reckoning (DR hereafter) on each terminal.

Thirdly, as considering about (3), we employed Allocated Topographical Zone (AtoZ hereafter) (Kawano and Yonekura, 2006). AtoZ is used in real-time peer-to peer type of VBG. By using AtoZ, each peer can decide strictly the ownership of the shared objects without synchronizing between peers about the ownership.

3 SYSTEM CONCEPT

In this section, we propose a new concept to design the real-time Web game.

3.1 Information Extraction Layer

In this section, we propose a new concept of hierarchical structure model (Figure 1). In Figure 1, Information Extraction Layer (IEL hereafter) is a new layer in the model and extracts value from information to be useful for the application or the person. For example, there are Google search engine that extracts search result from Web sites of the world, and access analysis tool that extracts access statistics in your Web site from access logs. In this paper, DR for prediction of remote avatar's attributes and AtoZ for ownership determination of the shared objects are shown as functions of IEL. Therefore, IEL mediates network and application (contents).

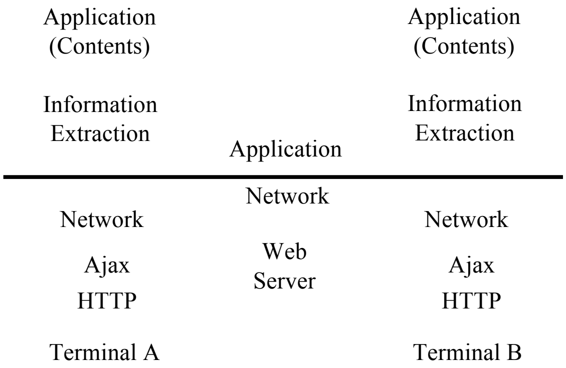

Figure 1: Hierarchical structure model.

3.2 System Architecture

Figure 2 shows the system architecture to satisfy the requirements in section 2. Figure 2 describes where system architecture was overlaid with hierarchical structure model in Figure 1.

In Figure 2, each terminal has five components, that is, GUI Handler, Modeler, Viewer, Virtual Space Resolver (VSR hereafter), and Network Handler. Here, MVC loop is performed on application layer in each client side. Moreover, interval of a MVC loop is asynchronous with that of network transmission with Ajax (Figure 3). In Figure 3, broken circle and arrow show transmitting. And from Figure 3, local terminal acquires the information of remote terminal in interval of a MVC loop, based on the data received in the interval of network transmission.

Then, we focus on roles of IEL, implemented the VSR in this paper. IEL mediates the Application (the GUI Handler and the Modeler) and the Network Handler, and manages the virtual space for the requirements in section 2.

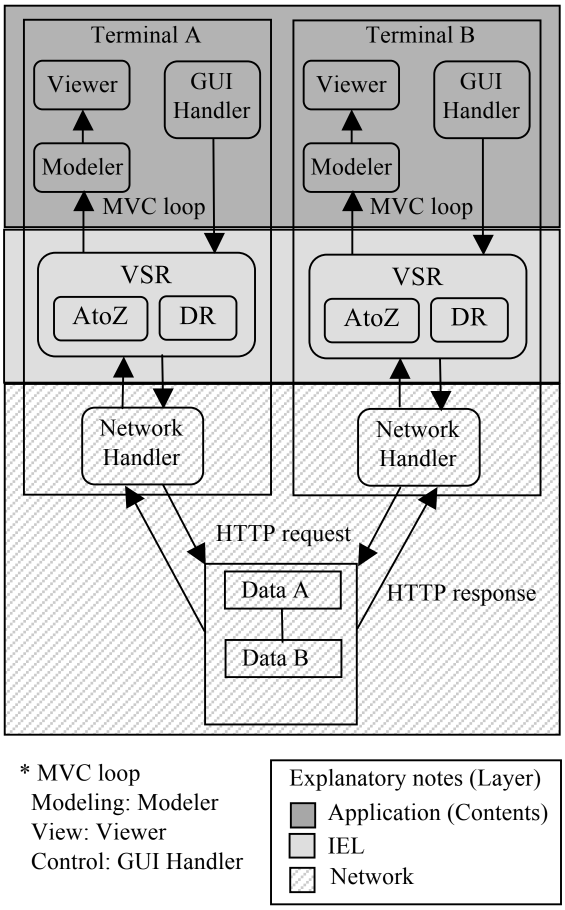

Figure 2: System architecture

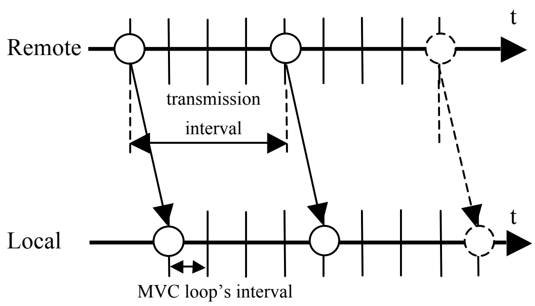

Figure 3: The time-chart of MVC loop and network transmission.

3.3 System Components

Here, we describe the role of Web server and each component that organizes the MVC loop.

3.3.1 Web server

The Web server performed as the media among terminals. Here, the data is used to acquire a virtual space, ex. a position of an avatar and so on, is transmitted using JavaScript Object Notation (JSON; The Internet Engineering Task Force, 2006) format instead of XML format, because of saving data size and parsing time (James Ward, 2007).

Web server receives the data of an avatar on each terminal at the timing of HTTP request. At that time, the server combines the latest data of each terminal like a chain, and sends it as HTTP response.

3.3.2 GUI Handler

The GUI Handler sends the user input to the VSR. User input is computed appropriately by the VSR, and provided to the Modeler.

3.3.3 Modeler

The Modeler receives appropriate information from the VSR. And then, the Modeler acquires the virtual space based on the information, and provides the virtual space to the Viewer.

3.3.4 Viewer

The Viewer presents the information of the game provided by the Modeler to a user. For this part, we employ Web browser for the Viewer.

3.3.5 Virtual Space Resolver

The Virtual Space Resolver (VSR) includes DR and AtoZ, and resolves the issues about virtual space (avatars and shared objects) in section 2. Here, the VSR receives user's operation, and provides the appropriate information to the Modeler. On the other hand, the VRS receives latest data about remote avatars from the Network Handler independent of the transmission above. The VSR is core component in this system.

There exists time interval until the data are received from the Network Handler. This time interval has some influences on the smoothness of the game progression. To avoid this influence, the data of remote avatars are extrapolated by DR (Singhal, 1996) based on the data with time stamp. DR is also a method used in networked computer games and simulations to reduce lag caused by network latency and bandwidth issues. Then, DR is applied to realize real-time Web games.

In this system architecture, synchronization regarding the ownership of the shared objects between terminals is not easily available because the architecture has no server for synchronization management. To solve this issue, shown in requirements (3) of section 2, we employed AtoZ as function of the VSR.

Thus, the VSR has the objective to provide appropriate information to the Modeler with no need for game developer to consider the influence of the network latency. Accordingly, the game developer can realize the Real-time Web game only by creating content like a stand alone program.

3.3.6 Network Handler

The Network Handler takes in charge of the transmission part to the Web server, using Ajax.

3.5 Advantages of the Design

Advantages of proposed design are as follow;

(1) Distributed game computing
(2) Development of only game content
(3) Usage of mash up technique

From (1), the construction of virtual space, which has been computed on the server in current Web system, can be achieved on each terminal and the load of the server is reduced. And then real-time Web game is realized. For (2), the VSR and the Network Handler are made to a component, and it is provided as API. Then Web game developer has only to create a game content. For (3), by using the technique of mash up and the combination with other site's API, it is possible to provide new services that create new additional values.

4 SYSTEM DESIGN

In this section, we describe concrete system design.

4.1 System Design Strategy

Purpose of system design is to implement our technology in the DVE for the real-time Web games. Concretely, we have already proposed the new DR protocol (Hanawa and Yonekura, 2007) and AtoZ (Kawano and Yonekura, 2006), and in this paper, these techniques are implemented.

4.2 Dead Reckoning Protocol

In this section, we describe the new DR protocol. The protocol has some advantages. In this section, we employed the protocol to the real-time web game (Figure 4). Figure 4 describes the details of DR component in Figure 2. In Figure 4, hierarchical structure (Application and the IEL) is shown by solid line, and three components (the VSR, the GUI Handler and the Modeler) are shown by broken line.

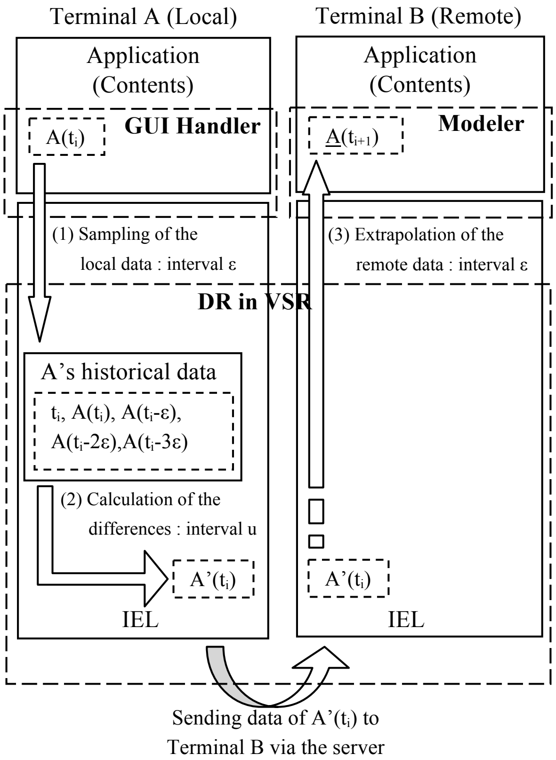

Figure 4: The dead reckoning protocol in our system.

Where $A(t_i)$ and $A'(t_i)$ indicate the position of avatar A at time t_i and the differences of avatar A at time t_i, respectively. Moreover, $\underline{A}(t_{i+1})$ indicates the predicted position of avatar A at time t_{i+1}.

The DR protocol in our system has the following three steps (Figure 4).

(1) Sampling of the local data

First, local terminal (Terminal A in Figure 4) records the data of local avatar in the interval of ε. And, ε is a very small interval time. And then, the data is updated to the own historical data. Where, the sampling data format is shown in Figure 5a.

(2) Calculation of the differences

Second, local terminal calculates the differences based on the historical data, these data are transmitted, in the interval of *u*, to the remote terminal (Terminal B in Figure 4). And, the transmitted data format is shown in Figure 5b.

(3) Extrapolation of the remote data

Third, remote terminal extrapolates the data of local avatar in the interval of ε.

In the above, local sampling interval ε is set less than the update interval *u*. Where, the DR protocol is utilized as on function of the VSR.

{ "name" : "name of the avatar (= ID)",
"Px ": x component of the position at time t,
"Py" : y component of the position at time t,
"Pz" : z component of the position at time t,
"t" : current time }

a. Sampling data format

{ "name" : "name of the avatar (= ID)",
"Px" : x component of the position at time t,
"Py" : y component of the position at time t,
"Pz" : z component of the position at time t,
"Vx" : x component of the velocity at time t,
"Vy" : y component of the velocity at time t,
"Vz" : z component of the velocity at time t,
"Ax" : x component of the acceleration at time t,
"Ay" : y component of the acceleration at time t,
"Az" : z component of the acceleration at time t,
"t" : current time }

b. Transmitting data format

Figure 5: JSON format used with dead reckoning protocol.

4.3 AtoZ (Allocated Topographical Zone)

In our previous works, we already proposed distributed processing protocol called AtoZ, which is used for P2P type VBG (Kawano and Yonekura, 2006). The VBG by using AtoZ is introduced on the Web. In MVC loop in Figure 2, AtoZ component is included in one function of the VSR like as the DR protocol in previous section.

4.3.1 P2P Type Virtual Ball-Game

In P2P type VBG, each avatar and shared object have the physical attributes (location, orientation and velocity). Then, a significant issue is how the ownership (decision of which peer owns the key to control a shared object) and the attributes for all avatars and shared objects should be kept consistent among peers. And, a player, or peer, can manipulate exclusively the avatar dedicated to that peer (no other player can manipulate that avatar). Each of the shared objects can always be continuously under the control of a certain avatar that is manipulated by a certain player. In other words, only one avatar exclusively owns the shared object at a time. The decision-making regarding ownership is constantly computed, and the ownership is dynamically transferred from one avatar to another.

4.3.2 Introduce of AtoZ

Under such conditions, we already proposed the following distributed cooperative protocols among the peers, regarding the ownership of a shared object.

At first, let's consider that each avatar has a priority field in which the ownership is given to that avatar while the shared object remains within the field. The priority field is defined as a set of spatial points in order to decide which avatar can gain access privilege to the shared object, and this must be uniquely decided among the avatars. Hence the priority field is decided dynamically, considering the relationship between the associated avatars in terms of their position, velocity, acceleration and orientation. In this paper, the priority field is referred to as the AtoZ formulated as

$$AtoZ(p_i) = \{\boldsymbol{x} \mid \boldsymbol{x} \in Z, Access(\boldsymbol{x}, p_i) = \min_{j \in N}(Access(\boldsymbol{x}, p_j))\} \quad (1)$$

Where p_i denotes the i'th avatar (i = 1,2,…,N).

In the formula above, AtoZ(p_i) is the AtoZ of avatar p_i, **x** and Z denote a spatial point and the whole space in the DVE, and $Access(\mathbf{x}, p_i)$ denotes the estimated elapsed time for avatar p_i, to reach out x. Thus, the AtoZ(p_i) indicates the set of points that avatar p_i can reach out faster than any other avatar. The diagram of the AtoZ is shown in Figure 6.

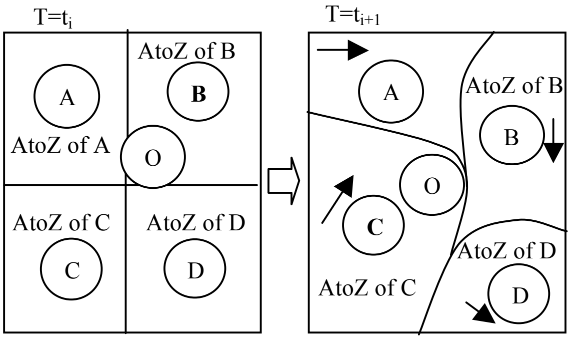

Figure 6: Overview of AtoZ.

If we assume that each peer uniquely determines the attributes of each avatar, each peer also commonly computes the AtoZs for all avatars, because they use the same algorithm to determine the AtoZ (i.e., priority field) with the same calculation accuracy. Thus, every peer may, though independently, select the same avatar as having ownership of the shared object because all the peers' decision making rule regarding ownership is the same. Then, this satisfies the consistency for the ownership.

4.4 Example of the Application

As a prototype system, an avatar operation game is implemented. In this system, a user logs in to the web site, the user can operate own avatar, and motion of his/her avatars is displayed on the browser (Figure 7). Figure 7 is an overview of the system. Figure 8 is screenshot of the application. In this application, we set interval of a MVC loop and that of network transmission to 100 ms and 300 ms respectively. And, URL of this Web site is as follows.

http://yard.cis.ibaraki.ac.jp/webgame/login.jsp

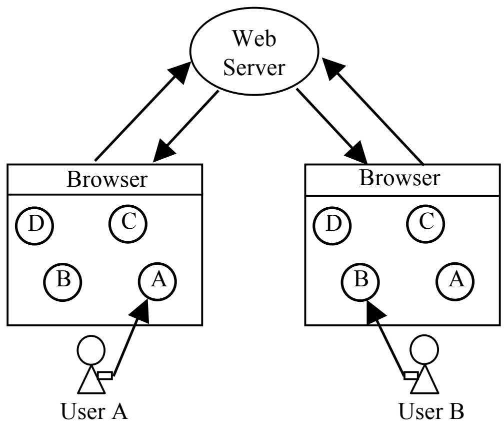

Figure 7: The system overview of avatar operation game.

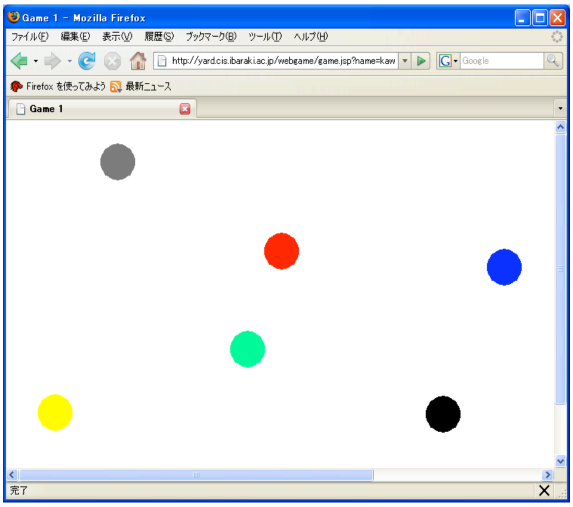

Figure 8: Screenshot of the application.

5 CONCLUSIONS

In this paper, we propose a new concept to design the real-time Web game. And, DR and AtoZ are what we employed in the real-time Web games to overcome limitation of HTTP communication. As an example of the application, Web based real-time avatar operation game is implemented. As a result, our concept may sufficiently expand to the various types of the real-time Web applications.

Our future works are enhancement of Web game with our concept, evaluation of validity about throughout of the game. Moreover, we will consider examples of application in addition to Web game and systematize our concept.

REFERENCES

Takemura, H., 1999. *Journal of the Virtual Reality Society of Japan,* vol.4, no.1, pp.59-63, (in Japanese).

Hashimoto, T., Sheridan, T.B., Noyes, M.V., 1986. " Effects of predicted information in teleoperation with time delay, "*Japan Ergonomics Society*, vol.22, no.2, pp.91-92, (in Japanese).

Valve Corporation, 2007. Counter-Strike, http://www.steamgames.com/v/index.php?area=game &AppId=240

Microsoft Corporation, 2007. Age of Empires III, http://www.ageofempires3.com/

KOEI Co., Ltd., 2007. SHIN-SANGOKU MUSOU, http://www.musou-bb.jp/

Motion-Twin, 2005. Dinoparc, http://www.dinoparc.com/

OpenAjax Alliance, 2007. http://www.openajax.org/index.html

The Internet Engineering Task Force, 2006, RFC4627: The application/json Media Type for JavaScript Object Notation (JSON), http://www.ietf.org/rfc/rfc4627.txt

Singhal, S.K., 1996. Effective remote modeling in large-scale distributed simulation and visualization environments, *Ph.D. Dissertation, Department of Computer Science,* Stanford University, Palo Alto.

Kawano, Y., Yonekura, T., 2006. Count Down Protocol: Asynchronous Consistent Protocol in P2P Virtual Ball Game for Multi-Player, *in Proc. ACM NetGames2006*.

Hanawa, D., Yonekura, T., 2007. Improvement on the accuracy of the polynomial form extrapolation model in distributed virtual environment, Springer-Verlag, Visual Comput, 23: pp.369-379.

Maid Marian Entertainment, 2004. Tank Ball2, http://maidmarian.com/Tank.htm

Jagex Ltd., 2007. RuneScape, http://www.runescape.com/

James Ward, 2007. Census - RIA Data Loading Benchmarks, http://www.jamesward.org/census/