A MAPE Loop Control Pattern for Heterogeneous Client/Server Online Games

Satoru Yamagata¹, Hiroyuki Nakagawa², Yuichi Sei¹, Yasuyuki Tahara¹, Akihiko Ohsuga¹
¹Graduate School of Information Systems, The University of Electro-Communications
Tokyo, Japan
{s-yamagata, sei, tahara, ohsuga} @is.uec.ac.jp
²Graduate School of Information Science and Technology, Osaka University
Osaka, Japan
nakagawa@ist.osaka-u.ac.jp

Abstract—Online games are recently gained attention according to the increase of the bandwidth of the Internet. On the other hand, when we develop an online game, we have to consider the time lag problem caused by the heterogeneous environments. In this poster paper, we try to apply MAPE loop model, which consists of four key activities (Monitoring, Analysis, Planning, and Execution) for adaptation, to help reduce the problem in the heterogeneous online game environment. In particular, we propose a MAPE loop control pattern for the heterogeneous client/server model. We experimentally embed the pattern into an online game application, and the results demonstrate that our MAPE loop control pattern will help to reduce the time lag problem in the online game application.

Keywords- client/server online game; self-adaptive systems; MAPE loop control pattern; time lag

I. INTRODUCTION

The number of online games and their users are rapidly increasing. They provide rich and up-to-date contents; however, they require real-time processing and fast interactivity. The time lag problem by network communication delay is one of the most serious problems. The quality of an online game goes down under the situation where the time lag occurs. In particular, match-type fighting game and FPS (First Person shooter) game require fast reaction speed and operation accuracy of game player in real-time. This paper focuses on inversion of an order according to the difference of the time lag of individual client. Moreover, online games need high permanency. Consequently, online game systems require that server applications receive much attention because of their ability to cope with the changes of environment, failures, and unanticipated event. The systems with such capability are called self-adaptive systems [1][3]. Self-adaptive systems should have certain decision mechanisms and using control loops is one of the approaches. The MAPE loop is one of control loops, which consists of four key activities (Monitoring, Analysis, Planning, and Execution) for adaptation [2][5][6]. Self-adaptive systems include these components as subsystems. However, practical systems including these components still have many problems, such as those design methods, domain characteristics and few actual adaptation examples [2].

In this paper, we propose a novel MAPE loop control pattern that can reduce the time lag influence. This pattern can be applied to the client/server type systems.

II. PROPOSED PATTERN

We propose a MAPE loop control pattern for client/server type systems, such as client/server online games (Figure 1). In this pattern, we deploy a Monitor and analyze component collects the performance data of the client, and the Analyze component analyzes the current situation the client. We also deploy a Plan and Execute components in the server, so that the server can determine and change to a suitable behavior at runtime.

A. The outline of a proposal pattern

The bottom of Figure 1 illustrates an instance of our MAPE loop pattern with four clients. We can apply this pattern to client/server applications that requires high accuracy in real-time but are deployed in heterogeneous environment. Heterogeneous environment means a distributed environment where the performance of hardware and software, or network bandwidth is different. In such environment, the next situation of whole system is hard to predict. Therefore, it is not desirable to deploy Analyze component to server. This pattern is for the applications that perform complicated processing and the
server manages individual clients. Therefore, it is desirable that the server grasps the analysis result of each client so that, the server changes the system configuration and behavior. From these points, our pattern deploys Monitor and Analyze components in individual client nodes, and deploys Plan and Execute components in the server node. As a result, the Monitor components can acquire individual clients’ data and the Analyze components can analyze the current situation of individual clients, which is difficult to recognize at the server side at runtime.

### B. The self-adapting scenario to a time lag of an online game

This pattern is for the applications that perform complicated processing and the server manages individual clients. When the server can update all clients’ data all at once and the clients can reflect the updated result in their side, the game keeps valid state. In this scenario, the server appropriately updates the updating cycle in consideration of the communication delay of individual client. When the delays of the client A and the client B are largely different, the server changes the updating cycle from \( w \) to \( w' \) (\( w < w' \)). This change enables the server to receive more clients’ data in the same cycle (Figure 2). Monitor component in individual clients collects individual network delays. Based on the collected data, the Analyze components analyze whether inconsistency. Based on the analysis result, Plan component in the server determines whether the updating cycle should be changed. Execute component changes the updating cycle according to the determination.

### III. EVALUATION

We experimentally implemented a virtual online game environment in Java and applied our MAPE loop pattern on application. In order to easily change delays of individual clients, we implemented the environment in a desktop PC. The delays are realized by using sleep method. In this environment, each client sends the selected command to server. The server updates to the current status based on the sent data from clients and send update status to the clients at the end of the updating cycle. In this experiments, we count inconsistencies that are occurred when there exists unreachable client data, which caused by the large difference of time lag among clients. We prepared two (experiment 1) and three (experiment 2) clients. In each experiment, we compared two applications; one was implemented without our MAPE loop pattern and the other included the pattern. Table 1 shows the experimental results. The results demonstrate that our MAPE loop pattern is effective to reduce the inconsistencies by suitably control updating cycles.

### IV. CONCLUSION AND FUTURE WORK

We proposed a MAPE loop control pattern for the client-server applications in heterogeneous environments. In particular, we focused on online game applications and their time lag problems. We have experimentally evaluated the effectiveness of our MAPE loop pattern through the experiments in a virtual online game environment. In future work, we try to improve the accuracy of the updating cycle by taking into account the prediction of environment by introducing a machine learning mechanism. We also plan to introduce a programming framework that helps developers to embed the MAPE loop pattern into existing online game applications.

### REFERENCES


