International Journal of Informatics Society

06/18 Vol.10 No.1 ISSN 1883-4566

Informatics Society

Aims and Scope

The purpose of this journal is to provide an open forum to publish high quality research papers in the areas of informatics and related fields to promote the exchange of research ideas, experiences and results.

Informatics is the systematic study of Information and the application of research methods to study Information systems and services. It deals primarily with human aspects of information, such as its quality and value as a resource. Informatics also referred to as Information science, studies the structure, algorithms, behavior, and interactions of natural and artificial systems that store, process, access and communicate information. It also develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields. The advent of computers, its ubiquity and ease to use has led to the study of informatics that has computational, cognitive and social aspects, including study of the social impact of information technologies.

The characteristic of informatics' context is amalgamation of technologies. For creating an informatics product, it is necessary to integrate many technologies, such as mathematics, linguistics, engineering and other emerging new fields.

Guest Editor's Message

Katsuhiko Kaji

Guest Editor of Twenty-eighth Issue of International Journal of Informatics Society

We are delighted to have the twenty-eighth issue of the International Journal of Informatics Society (IJIS) published. This issue includes selected papers from the Ninth International Workshop on Informatics (IWIN2017), which was held at Zagreb, Croatia, Sept. 3-6, 2017. The workshop was the eleventh event for the Informatics Society, and was intended to bring together researchers and practitioners to share and exchange their experiences, discuss challenges and present original ideas in all aspects of informatics and computer networks. In the workshop 33 papers were presented in eight technical sessions. The workshop was successfully finished with precious experiences provided to the participants. It highlighted the latest research results in the area of informatics and its applications that include networking, mobile ubiquitous systems, data analytics, business systems, education systems, design methodology, intelligent systems, groupware and social systems.

Each paper submitted IWIN2017 was reviewed in terms of technical content, scientific rigor, novelty, originality and quality of presentation by at least two reviewers. Through those reviews 16 papers were selected for publication candidates of IJIS Journal, and they were further reviewed as a Journal paper. We have two categories of IJIS papers, Regular papers and Industrial papers, each of which were reviewed from the different points of view. This volume includes five papers among the accepted papers, which have been improved through the workshop discussion and the reviewers' comments.

We publish the journal in print as well as in an electronic form over the Internet. We hope that the issue would be of interest to many researchers as well as engineers and practitioners over the world.

Katsuhiko Kaji received his Ph.D in information science from Nagoya University in 2007. He became a research associate at NTT Communication Science Laboratories in 2007 and an assistant professor in Nagoya University in 2010. Currently, he is an associate professor of Faculty of Information Science, Aichi Institute of Technology from 2015. His research interests include indoor positioning and remote interaction. He received IPSJ Outstanding Paper Award twice, DICOMO Best Paper Award twice, IPSJ Yamashita SIG Research Award for the second year in a row, etc. He is a member of IPSJ and JSSST.

Industrial Paper

Process Improvement Method Using a Non-model Approach

Akihiro Hayashi †and Nobuhiro Kataoka ‡

†The University of Shizuoka Institute of Science and Technology, Japan
‡Interprise Laboratory, Japan
pixysbrain@gmail.com, kataoka9@kataoka9.com

Abstract - Recently, the model approach-based process improvement method has been considered for improving the quality, cost, and delivery time (QCD) of system development projects. However, even when the capability maturity model integration (CMMI) maturity level 3 is attained, its actual effect has not yet been confirmed. On the other hand, the non-model approach identifies the process area that should be improved to achieve the specific goal of each organization's goal, and an actual effect is expected. However, the process improvement method of the non-model approach has not yet been formally established. In this paper, we propose a non-model approach method that has a real effect, such as QCD, and which is aligned to an organization's objectives. We apply this method to realize improvements in a real case process, and the results confirm the actual effectiveness with respect to the QCD.

Keywords: Non-Model Approach, Process Improvement, Defect Reports Analysis

1 INTRODUCTION

Owing to the prevalence of CMMI [1], which was developed in the United States, and the international standardization model of process assessment ISO/IEC 12207 [2] and/or 15504 [3], the process improvement for system development projects is gaining ground. A considerable number of companies are making the effort to introduce process improvement for their organizations. Consequently, the best approaches to process improvement have been widely discussed in Japan.

Process improvement is defined as the realization of a reduction for rework in system development, and an attempt to increase the QCD of system development projects.

There are two approaches to realizing actual process improvement: the model approach and non-model approach.

The model approach was introduced in CMMI, ISO/IEC 12207 and/or 15504, and aimed at achieving the conformance for the process management. The model approach includes process improvement approaches such as the CMMI's IDEAL approach and ISO/IEC 15504 ' s Geese approach.

However, the non-model approach does not adopt a process model but, instead, focuses on improving individual issues pertaining to each organization. In this approach, multiple processes are not discussed at the same time; however, unique processes are identified and improved to realize the business goal and foster the achievement of the organization. In the non-model approach, the total quality management (TQM), goal question metric (GQM), and plan-do-check-act (PDCA) approaches are known.

Model approach-based process improvement focuses on enhancing conformance to best practices such as CMMI. There are some cases that the activities do not substantially contribute to the improvement effect on business performance. On the contrary, non-model approach-based process improvements directly address points to be improved, and thus substantial improvement in performance can be expected.

However, no specific improvement approach has been established for non-model approaches. The outstanding issue rests on the nature of the non-model approach and how to proceed with procedures to establish an improvement approach.

According to the statistical information issued from CMMI Institute, 95% of companies adopting the model approach have completed improvements up to CMMI Level 3. In general, the CMMI appraisal is performed only once every three years. When an organization continues to conduct process improvements after achieving CMMI Level 3, the CMMI ' s appraisal report will not be referenced any more.

In the usual process improvement, we have conducted a very basic way of process improvement; namely, refer to the defect management reports of the development project, identify the area where the problem is concentrated, and then solve the cause of the upstream process that caused the problem in the downstream. We have confirmed some improvement through these basic activities. The motivation for this research is whether we could establish a process improvement method using a non-model approach by describing that method in a versatile manner.

Recently, there has been much focus on the model approach-based process improvement. However, approximately 30 years has passed since the software CMM Ver 1.0 was proposed, and yet there is often no real QCD improvement expected in the model approach. This is because in the model approach, the priority is sometimes on the activity to enhance conformance in order to achieve the maturity level. In this research, we aimed to improve the effect of QCD by focusing on improving performance using the non- model approach. When we applied the approach to the actual process improvement, the improvement effect was verified.

The procedure, we propose, first visualizes the magnitude of rework in each process by using the defect reports developed during the project life cycle. Next, we quantitatively identify the work products of the upstream process that have result in the largest rework. Next, we improve the process by adding the necessary practices to the process of development standard in the organization. When we apply the proposed method to an actual project and examine the effect for 3 years, the rework ratio, which was originally 51%, had improved to 42%. The organizational objective test defect ratio improved from 42% to 37%. We have got a prospect that this method is effective.

In a prior study in the field of software process improvement, Fukuyama et al. [4] reported that "what" is to improved is described in CMM, but the perspective of "how" it should be improved was missing. Then, they proposed a tool named SPIS (A Software Process Improvement Support System) that supports process improvement. However, this study focuses on the use of tools in the category of assessment.

Sakamoto et al. [5] analyzed the development process formally, and they quantified the effort of the amount of reduction achieved quantitatively, then showed the concrete benefit. Thereby, the method motivated the stakeholders of process improvement. In this study, they emphasized the improvement effects such as the reduction in the number of defects that are realized by improving the method of code review, unit tests, and joint tests on the development life cycle. However, it is not a proposal for process improvement in the sense of what improvement process can be obtained to improve the process in the non-model approach.

The authors of of this paper [6] proposed a method that identifies the process that needs to be improved by following the upstream process in accordance with the ISO/IEC 15504. The knowledge-management tools that formalize the knowledge, which was formalize-able but not formalized, originally pinpointed the process. We believe that this research is effective in that it identifies a process that includes the defects in individual organizations. However, in reality, few organizations utilize ISO/IEC 15504. Therefore, this is not a proposal for a process improvement method that can be used in general development projects.

Other studies that are related to process improvement have been reported. However, no prior studies have been reported that focus on the non-model approach, which identifies and improves processes that cause reworks by analyzing upstream defect reports ([7]- [11]).The methodologies for the non-model approaches have also been discussed as of the present; however, they are less practical compared to TQM, GQM and PDCA, which can be used in any field and not just process improvement. These methodologies do not focus on improving the QCD of the system development project. In non-model approaches, established methodologies similar to the IDEAL approach in CMMI have not yet been proposed. In this research, we propose a non-model approach-based methodology, which proves correct when applied to the actual process improvement site.

The remainder of this paper is organized as follows. In Section 2, we explain the concept and rework of the current process improvement method, after which we describe the issues to be solved. In Section 3, to solve the issues, we propose a method that visualizes the rework process by analyzing and improving the upstream defect reports. In Section 4, we explain a case study where this proposed method is applied to real-world organizations that have already achieved CMMI maturity level 3. In Section 5, we discuss the result of the case study, and we then conclude the paper in Section 6.

2 HYPOTHESES AND ISSUES OF THE REWORKED PROCESS

In this section, we present an overview of the current process-improvement activities and failure factors in the system development project. Then, we describe the hypothesis and issues to be solved using the non-model process-improvement approach.

2.1 Concept of Process Improvement Approach

Recently, model approaches such as CMMI have been used as approaches to realizing process improvements. The model approach is defined as an approach that improves the ability to manage processes by introducing the best practice model and improving the conformance to the process. As a proof of the effectiveness of the model approach, graphs are presented that show how the QCD has improved since the introduction of CMMI by organizations in accordance with level up maturity level 1-2-3 [12].

CMMI is an evaluation model of a system development project. It is the summary of the best practices used by the most excellent companies. Processes are evaluated in 5 stages. CMMI Maturity Level 3 is the level at which systematic process control is conducted. For organizations that have been improved up to Level 3, project groups under the control will manage projects using a common process consistently. Thus, the variation in quality and/or process will be less compared to Levels 1 and 2. CMMI was developed as a procurement model of the US Department of Defense. If any procurement projects occurred, the procurement section of an organization could expect that if they choose the organization that achieved CMMI Level 3 as the development partner, any project under that organization will demonstrate consistent performance. For this purpose, Level 3 is adopted as a procurement standard.

However, the model approach does not guarantee improvements in the QCD for organizations where CMMI and/or other approaches have been introduced. The graph presented as evidence that a process has improved is just one incident. More importantly, there are cases that report no specific improvement in the QCD effects for organizations in which CMMI was introduced and achieved maturity level 3 [13].

On the other hand, a non-model approach is defined as an approach that improves the ability to manage processes by improving the performance of individual processes required for each organization without using a given process as best practice. In the non-model approach, rather than improving the process from the beginning to the point at which the best practice is determined, it identifies the processes that are required to solve individual issues in each organization, and improves the processes to their ideal states.

Figure 1 shows the causes of failure and their ratio in system development projects [14]. According to this survey, 45% of the causes of failure in system development are related to

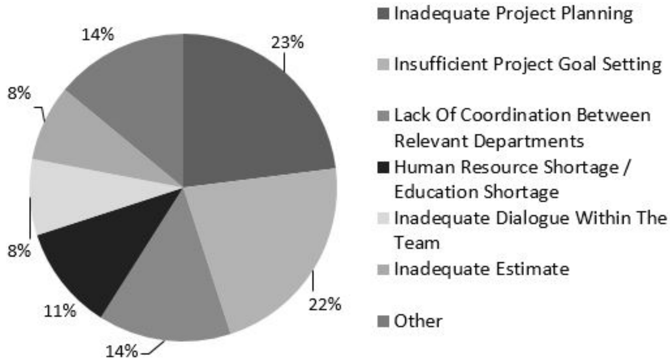


Figure 1: Causes of failure and their frequency of occurrence in system development projects

two factors: the inadequacy of the project plan and insufficient goal setting. Even if we consider other causes of failure, many of them are process problems that are caused by management. Given such a premise, in order to maximize effect of process improvement, we will make improvements with respect to the process area of "project planning," which is where most of the problems are concentrated.

However, even if the "project planning" process is improved until it has an ideal state, the QCD of the system development project will not necessarily improve. The reason for this is that the process improvement does not directly improve the QCD, but reduces the "rework" that was originally unnecessary. Then, this resulted in the improvement of the QCD.

Therefore, for process improvement that is realized using the non-model approach, we need to define the rework, and identify and improve the process that can reduce the need for the rework.

2.2 What Is a Rework?

In order to explain what is a rework,let us quote the project cost classification in system development proposed by Bill Curtis in Fig. 2 [14].

In system development, the project cost can be classified as an implementation cost and quality cost. The implementation cost is the total cost to manage a project, and includes the preparation of a project plan, progress management, reporting, and the engineering cost for manufacturing. Next, the quality cost is the cost to improve the quality of the product to be provided. Furthermore, quality costs are classified as compliance costs and noncompliance costs. The compliance cost is the sum of the costs related to evaluations, such as reviews, tests, and audits for quality-control purposes, and preventive cost such as training, procedures, tools, and data classification.

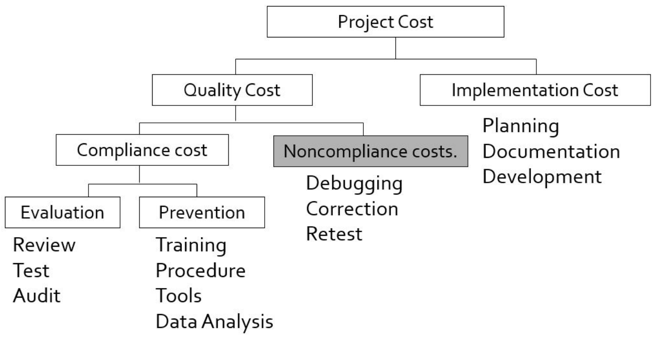


Figure 2: Classification of project costs

Meanwhile, noncompliance is the cost of debugging, correction, retesting, etc., which occur because of the cost of activities to ensure compliance. Because noncompliance costs correct defects that are noted in conmliance activities, there is no noncompliance cost if there is no noncompliance. According to Bill Curtis' research, as a result of investigating leading global companies such as IBM/TRW/NASA-SEL/HP/Raytheon, 30-40% of a project's cost is the noncompliance cost.

Even if the productivity itself is not expected to improve, the removal of this noncompliance cost by process improvement is expected to significantly improve the QCD. In this research, we define the work generated due to noncompliance in Fig. 2 as "rework."

2.3 Hypothesis of Process Improvement to Reduce Rework

A rework can be classified into two types: Noncompliance due to defects before the upstream process and noncompliance due to defects in the process. For example, when a defect is detected by a source-code review, if it was wrongly coded owing to the use of an incorrect design document, it becomes noncompliance due to a defect before the upstream process. When the correct design document is used, if it is an error in the code, it is a noncompliance that is due to a defect in the process. In order to reduce the reworks, it would be prudent to improve the upstream process that resulted in the noncompliance defect in downstream processes.

In this research, we focus on defect reports in order to reduce reworks due to defects in the upper stream process. The defect report describes defects that are detected during the process. The contents of the defect reports are mainly based on basic information such as the date, the person in charge, the defect registration number, and the target system. In addition, the defect detection process, the defect outline, the generation location, the treatment outline, and the defect incorporation process are included.

Here, if the defect-detection process and the mixing process coincide with each other, it is a noncompliance due to a defect in the current process. On the other hand, if the detection process and the mixing process do not match, it can be regarded as a defect caused by the upper stream process. The basis of process improvement is root-cause management, and improving the upstream process can result in greater benefit by improvements in the reduction of rework.

From this perspective, we hypothesized that it is possible to reduce the rework by tracing back to the upstream process that caused the defects using the defect reports; we can identify and improve the processes that are more beneficial to process improvement in the downstream process.

2.4 Issues to Be Solved

In order to establish a process-improvement method that is based on a non-model approach, the following two issues should be solved.

1. Establish procedures to identify processes that caused rework.

   The process-generating rework is different for each organization. We have not established the method to identify and improve the process of the upstream process that caused the rework in individual organizations.

2. Establish procedures that contribute to the realization of organizational objectives.

   An organization's expectation for process improvement should be to meet some organizational goals. In many cases, there may be an individual goal other than improving QCD which accompanies the reduction of rework. For this reason also, we chose the non-model approach. Procedures for process improvement to contribute to achieving such individual organizational objectives are not yet established

3 IMPLEMENTATION METHOD OF PROCESS IMPROVEMENT ACTIVITIES USING NON-MODEL APPROACH

In Section 3, to solve the issue described in Section 2.4, we propose the process improvement implementation procedure using the rework information of defect reports. The proposed method consists of five phases, the setting of organizational goals for process improvement, the visualization of the rework effort, the identification of work products in the upstream process to be improved, the root-cause analysis for embedding noncompliance, as well as additional practices to eliminate root causes.

Improvement approaches such as TQM, GQM and PDCA have been discussed even in non-model approaches. However, they are less practical to implement. The person in charge of process improvement do not know which process to choose and how to improve them; therefore, the method to be established ought not to be an abstract one but a "procedural" method. By improving the process according to the procedure, substantial performance improvement can be expected. For that purpose, it is desirable to adopt a method of naturally deriving the result by following the procedure.

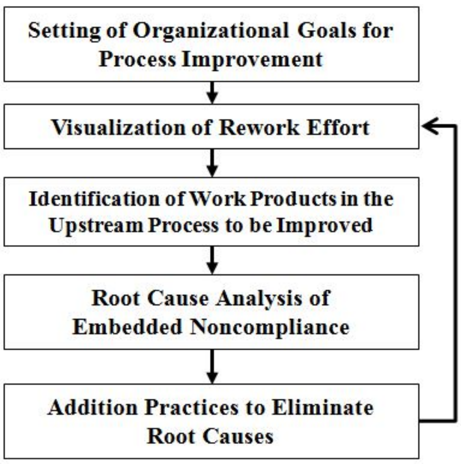


Figure 3: Implementation method for process improvement in non-model approach

In the model approach, as a result of improving the conformance to the model, the performance is expected to improve as well. However, by improving the conformance to the model, it is inevitable that unnecessary practices of that organization will be improved in the process. In this research, the proposal first identifies the process that caused the rework, and revises the work products in the upstream process that caused the defect to the process. Thus, it is possible to improve the necessary processes directly. This, however, does not mean that performance improves as a result of improved conformance as in the model approaches. It is only a very simple process improvement aimed at performance improvement.

3.1 Setting of Organizational Goals for Process Improvement

First, we set up organizational goals for process improvement. Organizational goals are to be achieved using process improvement. At the same time, an index to measure the organizational goal quantitatively should be determined.

Examples of organizational objectives include achieving a given maturity level, improving the quality and process performance objectives (QPPO), improving productivity, reducing market outflow defects, and so on.

The indexes used to measure the organizational objectives quantitatively are achievement requirements that assess whether organizational objectives have been achieved. If in the case where an organizational goal is to reduce the market outflow defects, the quantitative index should be "the ratio of the number of market outflow defects will be 10% lower than the previous year." The index should be a concrete expression such that the cycle of data collection and/or verification analysis could be carried out as instantaneously as possible.

3.2 Visualization of Rework Effort

Generally, if requirements that are related to QCD, such as the order value, quality requirement, and development time are advanced, they may influence the total project effort. Therefore, we calculate the effort ratio for each process in the whole project, and then we visualize the effort expended for each process. In this case, the effort is defined as the modified effort described in the defect reports.

3.3 Identification of Work Products in the Upstream Process to Be Improved

In the waterfall-development life cycle, work products that are created in a certain process become the input to the next process. Defects that occur in the downstream process are caused by defects in work products from the upstream process. For example, a requirement definition is considered a work product in the requirement-management process, and the requirement definition form becomes an input to the design process. The reason for the occurrence of some problems in the design process may be because of some element of trouble in the requirement-definition document.

We first pinpoint the upper stream process the defect contamination based on the amount of rework effort, which is described in Section 3.2. The work products made in each process are documented as a work product list at the time that the configuration-management plan is developed. These documents should include work products that caused defects in the downstream process. Considering the unique circumstances of each organization, work products that contain defects are identified logically.

3.4 Root Cause Analysis of Embedded Noncomliance

The reasons for which work products are incomplete are related to cases where the work products themselves were defective, where it is difficult to highlight the defects by peer review, and where the defects are not detected only by peer review. This may be the case where an intention in the upstream process is not transmitted correctly or when similar defects that had to be corrected at the same time are missed. This would be difficult to determine in the review. In such a case, we will follow the work products and processes that caused the upstream process.

3.5 Addition Practices to Eliminate Root Causes

In organizations in which process improvements are introduced, their activities are carried out according to the "organization's set of standard processes (OSSP)". The OSSP is classified in the process area, and is described as the process that produces a specific kind of work product. Therefore, we can identify a process that is performed according to the result of the work product.

By identifying work products that resulted in a defect in the downstream process, we can identify the process that is performed by examining the OSSP. Therefore, we can improve the process by adding practices to eliminate the cause of the defect embedded in the work product.

The above is the proposed method employed in this research. However, even if one practice is added to a process that has already been established and operated, it is difficult to have a notable effect. More importantly, to contribute to the achievement of organizational objectives, multiple process improvement methods should be simultaneously utilized.

In this research, if defects due to the influence from upstream process account for more than 10% of the defects, the cycle of 3.2 to 3.5 for this proposed method shall be continued.

4 APPLICATION EVALUATION

In Section 4, by performing a case study where the method proposed in Section 3 is applied to a real case of process-improve- ment activities, we can evaluate the effectiveness of the proposed method.

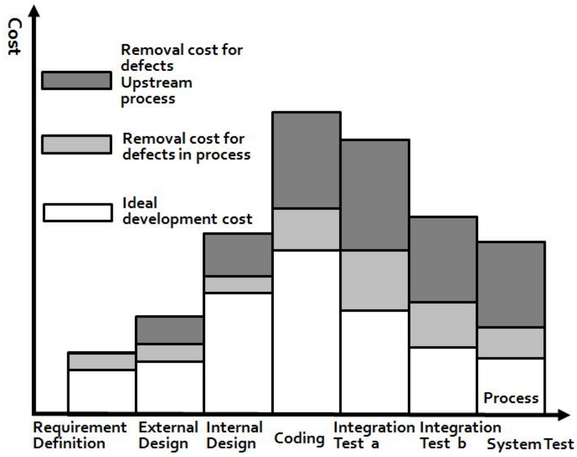


Figure 4: Development Lifecycle and Effort Ratio

4.1 The Target Case

Company A has been developing an embedded system mainly for the purpose of measurement equipment. They have been working on CMMI activities for 10 years, and achieved CMMI maturity level 3 four years ago. Although the process areas required for system development activities have been introduced in the organization, there is so far no concrete improvement in the QCD.

Company A adopts the waterfall model's development lifecycle, which consists of a requirements definition, external design, internal design, manufacturing, integration test A, integration test B, and system test. Integration testing is the testing of interfaces between components. It distinguishes integration test A, which performs to satisfy external specifications and internal specifications. Integration test B targets external specifications at interfaces between subsystems.

Company A has chosen this proposal method rather than CMMI high maturity in order to achieve effective process improvement after achieving CMMI maturity level 3 and developing the underlying process.

4.2 1st Round

The organization goal was set as "Left Shift by Reduction of Test Defect Ratio". The test defect ratio defines the ratio of the number of detected defects in the test process vs. the entire life cycle upon completion of the manufacturing process. Left shift is defined as an activity to prevent them from rework in the downstream process by detecting larger defects in the upstream process.

When Company A introduced this proposed method, the test defect ratio was 42%. The index used to measure the organizational objective quantitatively is defined such that "the Test Defect Ratio was Declined for Three Consecutive Years from the Start Point". Figure 4 shows the development life cycle used by Company A to describe the effort on the horizontal axis and each process on the vertical axis. The effort on the horizontal axis is defined as the ratio of the effort for each process in the total effort for the project.

Company A carries out a peer review in the upstream process against the requirement definition document, external design document, internal design document, and source code document. In the downstream process, the tests are conducted in integration test A and integration test B, and the system test and logs are respectively documented. The peer-review record and the test log are included in defect reports.

We calculate the rework effort from these defect reports, and in Fig. 4, we draw noncompliance costs for defects in the upstream process as dark, incompatible costs for defects in the current process as light, and the ideal development cost as white. The sum of the noncompliance costs for defects before the upstream process and the nonconformance cost for the current process was 51%.

When summing the defect reports, 23% of the dark color area was embedded in the external design process. Company A deals with custom-made measuring equipment, products which once paid for, are used for more than 10 years, and derivative development that is related to function expansion is repeated. In the external design process, a system architecture design document, a screen design document for software, and external and internal interface design documents are made. Among them, the architectural design documents are reused, and the screen design document is already agreed with the customer, so we suspect that the defect is embedded in the interface design document.

Next, in order to analyze the root cause of the incompatibility embedded in the interface design document, we conducted investigations from three points of view: 1) Input to the interface design document, 2) completeness of the interface design document itself, and 3) the absence of a peer review of the interface design document. The incompatibility between the interface document and other classified work products is significant.

In the development by Company A, there is a sub project manager (PM) who is in charge of electricity, machinery, and software. It is necessary to simultaneously modify the interface design document under the management of the sub PM when we receive a change request from customer. However, if the analysis of the scope of the impact due to the change is insufficient in some cases, some interface design documents will be missed. This causes defects in the downstream process. In other words, it was determined that the root cause of the trouble was the change-management process.

Company A also introduced a change-management process, but often changed the source code suddenly in response to change requests, and sometimes did not revise the interface design document. There was no hand to the so-called "skewer management," which simultaneously and consistently modifies the interface design document and source code.

In the OSSP of Company A, there is a practice of impact analysis, which is done by opening a configuration control board (CCB) at the time of change management. However, since CCB is not an essential practice in CMMI, Company A arbitrarily performs CCB. Therefore, Company A incorporated practices to perform CCBs based on the judgment of the PM and sub PM when a customer's requirements changed.

In addition, as there is a possibility that the impact analysis itself is inadequate even after performing the CCB, we also added practices to number the traceability matrix from the requirement definition document to the interface design document. If the traceability matrix is assigned a number, it can also be useful for analyzing the influence range, and it is possible to prevent omissions to corrections of similar parts in the downstream process.

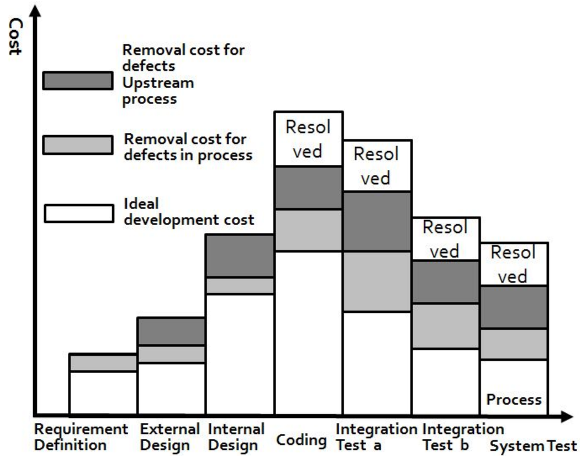


Figure 5: Development lifecycle and effort ratio (2nd)

4.3 2nd Round

Figure 5 is a graph that was created for the visualization of rework after the completion of the 1st round of process improvement proposed in this paper. Because the height of the bar graph signifies the effort ratio of each process, it is the same as in the 1st round. "Resolved" is the area that is expected to be resolved in the 1st round of the proposed method.

After we excluded the portion referred to as "resolved," when we compiled the defect reports for each process, 12% of the dark area was in the internal design process. This means that even in the second round, more than 10% of the defects were concentrated within a process. Therefore, we proceeded to the next step.

Software design documents and state-transition diagrams are primarily created in the internal design process. In the embedded development, the software design document is patterned beforehand, and defects other than careless mistakes are not often embedded. Therefore, it was inferred that there is a defect in the state-transition diagram. Based on an actual examination, defects were included in the interface between the newly developed part and the diverted part in the derivative development. When a state-transition diagram contains a defect, there should be a notification of a mistake when the state-transition diagram is developed. However, because it is difficult to complete a complicated state-transition diagram, we determined that the main reason of the defect was the absence of peer-review.

In OSSP of Company A, the implementation rate of the peer review was classified according to the criteria of the scale of work products. In this case, the peer-review implementation rate for documents such as design documents was 100%, and for transition diagrams and source codes, it was around

30%, centering on ”dangerous places” based on the PM judgment.

Therefore, in the peer review of the state-transition diagram, when the reviewer reached 30% the exit criteria, they should then terminate the peer review. For this reason, the part including the defect may have deviated from the peer-review target range.

In other words, 30% of the peer-review implementation-rate was not included in the range that the PM judged as ”a potentially dangerous place”, and that was the root cause. Therefore, a new practice was added such that the PM and sub PM must negotiate the validity of the peer-review scope and prepare the minutes of the meeting.

There are three peer review methods at company A: the buddy method, team method, and reading-out method. The buddy method is performed by two persons, a creator and reviewer. The team method is a method in which multiple reviewers conduct an individual preliminary review. The reading-out method is a walk-through method in conference format. It does not include a preliminary review, and the producer of the work reads the review subject at the review meeting, and explains the work product in order from the beginning to the end. The peer review becomes strict in this order, and instead of requiring many attempts, the defect detection rate increases.

Company A adopts the buddy method when anything is specified, but in the case of ”newly developed parts” that are in the ”dangerous place”, we added the practice to the review with the reading-out method.

4.4 3rd Round

Similarly, in the 3rd round, we created a graph corresponding to Fig. 5, and we visualized the classification of incompatibility using color coding. However, because the defects before the pre-process did not exceed 10%, we decided to end the cycle of this proposal.

4.5 Application Result

At Company A, a project will be launched each time a request occurs. There are relatively short-term projects with duration of about six months. After achieving CMMI maturity level 3, we applied the proposed method every year to improve the process. In this section, we evaluate the effectiveness of the proposed method by using the result of applications over the three-year period.

1. Have procedures been established to identify and improve the process that caused the rework?

   After confirming the organizational goal of process improvement, we established the process improvement procedure that first visualized the rework effort and identified the work product of the upstream process to be improved. Then, we identified the root cause of the defect by further analyzing to determine the cause of the process incompatibility, and we added the practice for process improvement. Although the proposed method focused on reworks that are due to upstream defects,

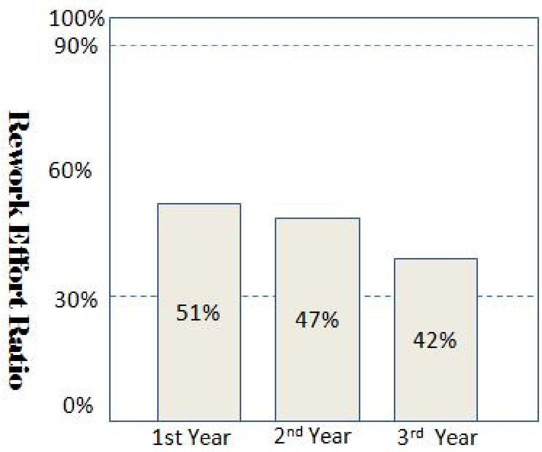


Figure 6: Rework effort ratio

   for comparison with the case where CMMI was introduced, it is necessary to include reworks due to defects of upstream processes and due to in-process defects, as in Fig. 6. The rework effort ratio, which was originally 51%, improved to 42%. Even though they achieved CMMI maturity level 3, for an organization that could not confirm the specific effect of the improvement, an improvement effect of approximately 9% was confirmed three years after introducing this proposed method.

2. Has an improvement procedure that contributed to realizing organizational objectives been established?

   In the proposed method, we set up organizational objectives at the initial stage, and at the same time, we set indices to quantitatively measure organizational objectives in order to see that process improvement contributes to the realization of organizational objectives. Figure 7 shows the trend of the test defect ratio over three years after introducing this proposed method. The test defect ratio improved from 42% to 37%. The number of projects in the first year after the introduction of this proposed method was 102, and the number of dprojects after three years was 96. Here, as the null hypothesis, we wet the ”Mother proportions of two groups are equal” and tested the difference between the ratios of the two groups. If the number of projects is a trial number, the improvement from 42% to 37%, the test statistic exceeds the p value. It can be said that the confirmed improvement is a significant effect.

5 DISCUSSION

In this section, we evaluate the validity of this proposal by analyzing the result of applying the proposed method of this research to the case study of chapter 4.

5.1 Effectiveness as a Non-model Approach Procedure

In the proposed method, as shown in Fig. 3, if the improvement policy is agreed first, the rest of procedure can follow the

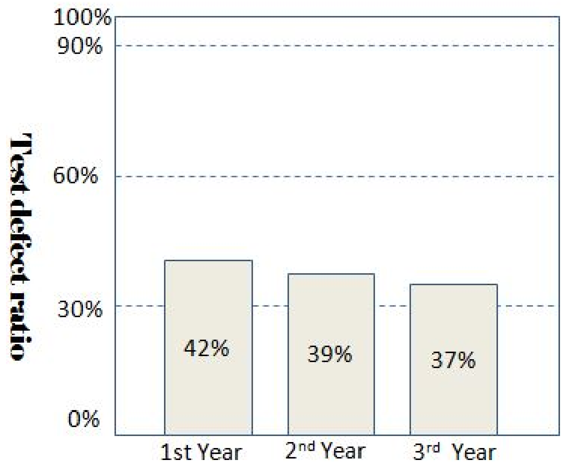


Figure 7: Test defect Ratio

improvement process according to the procedure and confirm the reproducibility.

Taking company A as an example, projects under the organization resulted in a unified form of the defect management reports in each process. By using the defect management reports, the rework efforts are then visualized as shown in Fig. 4, from which one can identify the areas with the most rework. Thereafter, by referring to the configuration management process, it is possible to uniquely identify the upstream process that caused the rework. We will carry out the peer review again and confirm the intent of the upstream process, from which one can identify the root cause the process in downstream. Finally, process improvement is achieved by adding practices to solve the root cause.

This methodology is not based on empirical results or implicit knowledge of process improvement. The same conclusion can be reproduced by tracing the upstream process according to the procedure. It is considered that the proposed method can be utilized as a non-model approach procedure.

5.2 Effectiveness as Proposed Method of Process Improvement

According to the case of company A, the rework rate was improved from 51% to 42% and the test defect ratio was improved from 42% to 37%

The rework rate improved by 9% over three years, which is only 2% lower than 40% of the leading company. Originally, CMMI was intended to be suitable for large-scale enterprises like the leading company. There has been some discussions that it may not be suitable for a small-scale enterprises such as company A. Improvement to the remaining 2% of the lower limit is thus considered acceptable.

The test defect ratio was improved by 5% over three years. This value was not expected from the beginning; it was obtained as a side effect owing to the reduction in the rework ratio. In the announcement for the SEPG conference to be held every year in North America, because there are many companies aiming for a test defect ratio of approximately 35%, the target value is again improved to 2% for average companies.

Because introducing one improvement method is a not a panacea for all problems,, both the rework ratio and the test defect ratio have to be improved to 2% up to the general target value. This may be acceptable as an improvement for the first three years.

5.3 Requirement for Introducing This Proposal Method

In order to implement the procedure of this proposed method, it is necessary that a certain degree of process has been established prior to that time. The proposed method worked effectively because it can be implemented practically, such as creating a unified defect management sheet, identifying work products created in each process, and managing work product output from each process. It is not possible to introduce anything without the foundation of a process management.

For company A, the improvement up to CMMI Level 3 was achieved, thus it can be said that the proposed method functioned effectively. However, one cannot expect that the proposed scheme will work similarly where the process is not introduced like the organization at CMMI Level 1.

If a company that achieved CMMI Level 3 desires to continue with further process improvement, it is natural for the company to aim for Level 4 or 5, which is a high-maturity level. However, if organizations are wary of improvements based on the model approach, then there is the option to introduce improvements using this proposed method, which is based on the non-model approach.

6 CONCLUSION

While many process-improvement activities are based on model approaches, in this paper, we proposed a process improvement method based on a non-model approach that appears to contribute to the achievement of each organization's business goals.

The company A that was referred to in this paper did not confirm the concrete effect while attaining the CMMI maturity level, but in the three-year period after adopting this proposed method, we confirmed the effect of realizing the goal and reduction in the number of reworks. We therefore believe that this proposed method is effective.

To further improve the proposed method, it is necessary to change some subjective judgments in the procedure to make them more objective. The defect reports are sometimes created during the project activity, the embedded upstream process defects are "surmised". Even when multiple factors causing defects are considered, it is sometimes difficult to express them accurately in defect reports.

REFERENCES

[1] SEI, CMMI for DEV Ver1.3, SEI, CMU,(2011).
[2] ISO/IEC 12207, "Information Technology - Software life cycle processes," JIS X 0160,(2012)(in Japanese).
[3] Software Process Seminar, "Receiving the standard announcement of TR," Japan Standards Association, In-

formation Technology Standardization Center,(2000)(in Japanese).
[4] S.Fukuyama, S.Miyamura, H.Takagi, R.Tanaka, “A Software Process Improvement Support System SPIS,” IEICE Trans. Inf. Syst.,Vol.E83-D No.4 pp.747-756,(2000)(in Japanese).
[5] K.Sakamoto, T.Tanaka, S.Kusumoto, K.Matsumoto and T.Kikuno, “An Improvement of Software Process Based on Benefit Estimation,” In the Trans of the IEICE. pp.740-748,(2000)(in Japanese).
[6] A.Hayashi and N.Kataoka, “Identification Method in Software Process Improvement Areas,” IEEJ Transactions on Electronics, Information and Systems 128(7), pp.1231-1241,(2008)(in Japanese).
[7] H.Tuan, C.Liu, and C.Chen, “Using ABC Model for Software Process Improvement: A Balanced Per-Spective,” Proceedings of the 39th Annual Hawaii International Conference on System Sciences(HICSS’06)(2006).
[8] L.Mathiassen, O.Ngwenyama, and I.Aaen, “Managing Change in Software Process Improvement,” IEEE Software,pp.84-91,(2005).
[9] D.Kozlov, “Evaluation and Improvement of the Software and Process Quality through Their Cross-Impact,” 11th IEEE Int’l Software Metrics Symposium (METRICS’05), pp.43,(2005).
[10] B.Freimut, C.Denger, and M.Ketterer, ”An Industrial Case Study of Implementing and Validating Defect Classification for Process Improvement and Quality Management, ” pp.19, 11th IEEE Int’l Software Metrics Symposium (METRICS’05),(2005).
[11] D.Caivano, “Continuous Software Process Improvement through Statistical Process Control,” Ninth European Conf on Software Maintenance and Reengineering (CSMR’05), pp.288-293,(2005).
[12] J.Herbsleb, D.Zubrow, Dennis Goldenson, Will Hayes, Mark Paulk, “Software quality and the Capability Maturity Model,” Communications of the ACM, Volume 40 Issue 6, pp.30-40,(1997).
[13] J.Herbsleb,A.Carleton, J.Rozum, J.Siegel, and D.Zubrow, “Benefits Of CMM-Based Software Process Improvement: Initial Results,” Software Engineering Institute, CMU/SEI-94-TR-13,(1994).
[14] A.Hayashi, “Process Improvement Method by Non-Model Approach,” IEICE, SWIM2016-13, vol. 116, no. 341, pp.11-18,(2016)(in Japanese).

(Received September 25, 2017)
(Revised December 26, 2017)

Akihiro Hayashi received an MBA and a Ph.D degree from the University of Tsukuba, Tokyo, Japan. His professional career includes International excellent companies: Motorola, NTT, and IBM. He has become a Professor of the University of Shizuoka Institute of Science and Technology in 2018. His research interests include Total Quality Management, Process Improvement, and Quantitative Project Management.

Nobuhiro Kataoka received a Ph.D degree from the University of Tohoku, Japan. He joined Mitsubishi Electric Corp. in 1968 and became a technology director at Information System Technology Center in 1997 and a Professor of Tokai University in 2000-2009. His research interests include Quality Control, Business Process Modeling, Business Modelization Methodologies. He is now a part-time instructor of Tokyo Denki University. He is also a member of IPSJ and a fellow of IEICE.

Regular Paper

Automatic Motion Capture System of Actors' Body Movements with Action Cueing

Takayoshi Takano[*], Hiroshi Shigeno[**], and Ken-ichi Okada[**]

[*]Graduate School of Science and Technology, Keio University, Japan
[**]Faculty of Science and Technology, Keio University, Japan
{takano, shigeno, okada}@mos.ics.keio.ac.jp

Abstract -The body movement of an actor is one of the important factors in theatrical performance. In theatrical practice, a director instructs actors about such important element. In addition to practice with the whole members, actors practice individually to check and review the director's instructions. Therefore, to keep a record of practice is important to improve the quality of performances. Currently, the recording of practice is done mainly in the form of writing to the script by text, but it is sometimes hard to record the body movements accurately. Therefore, we constructed a system aimed at automatic motion capture of actors' body movements in theatrical practice with action cueing. By notifying the timing of performance for an actor using smart watch, our system can record the physical expression while supporting the actors' practice. We conducted an experiment to evaluate the efficiency of this system. From the results, we confirmed that this system can record the body movements with a small workload and short time.

Keywords: motion capture, theatrical performance, body movement, smart watch, action cueing

1 INTRODUCTION

Theatrical performances are created by directors, actors and other many staffs. In theatrical performance practice, the director instructs actors about performances. Theater organizations practice many times to make the performances better. However, many theatrical organizations have problems of lack of practice places and opportunities [1]. Therefore they encourage actors' self-practicing to improve the quality of performances. In self-practice, actors mainly review the director's instructions and prepare for the roles by reading the script for the next practice. Therefore, keeping a record of practice is very important to improve the quality of the performance. Currently, the recording of practice is done mainly in the form of writing to the script by text, but it is difficult to record the body movements of actors by text. We focus motion capture technology as a method to record body movement to deal with this difficulty. However, using motion capture in a limited practice time may be a heavy burden for the director and decrease practice time.

In this paper, we construct a system aimed at automatic recording of actors' body movements in theatrical practice while supporting actors' learning. In recording practice, it is difficult to grasp the timings of actors' performances. To deal with this problem, we utilize action cues for supporting theatrical practice. We assume that action cues are not only useful for supporting theatrical practice but also for grasp timings of actors' performances. By automating recording body movements of actors', it is unnecessary to devote time for motion capture separately from the practice time, and it is possible to reduce the workload of recording body movements.

In order to evaluate user's workload of this system, we conduct experiments in two situations: recording by this system and by a human-shaped input device. In both situations, the subjects participate as the director and record the physical expression of actors. Two experimenters participate as actors. In the experiments, we compared workload and time required to record actors' body movements. From the results of the experiments, we confirmed that this system can record body movements by lower workload and shorter time. The rest of this paper is organized follows: Section 2 describes overview of theater activities. Section 3 introduces Digital-Script, and Section 4 describes related works. Then, the detail of our system is described in Section 5, Section 6 shows evaluation of this system. Finally, we conclude in Section 7.

2 THEATER ACTIVITIES

This section describes the outline of theatrical practice and the methods of recording practice.

2.1 Outline of Theatrical Practice

In Japan, theater activities are performed by many organizations continually since early time, and it is recognized as an important artistic activity. When performing theaters in public, a theater organization completes the work while practicing repeatedly many times. A director's role by the process of the practice is to coach actors to bring their performance close to ideal. On the other hand, actors try to understand the director's intention and express it in the performance. Steps of a practice are roughly divided in three, “Reading Scripts”, “Standing Practice”, and “Stage Rehearsal” [2]. According to Goan et al., in these practice steps, “Timing of each action”, “Body movements”, and “Feelings” are important [3]. A director mainly instructs actors about these important elements during practice.

- Timing of each actions

Timing of the start of a line creates a span between lines. Different from a picture work such as movies, spans are transmitted to the audience in real time. Spans are the time the audience uses imagination about the kind of relationship between characters. Therefore, spans are very important element being concerned with success or failure of performances [4]. In actual practice, instructions from a director to an actor about spans are the most frequent. A director coaches actors about it very finely. As an example, "a little early", "immediately begin to say when the line of that person is finished", "open for two seconds." are given. By taking the appropriate spans, performances become better, and it is possible to avoid lines become monotonous.

- Body movements

The actors express the feelings of the roles playing with their own bodies. Frequently, the director instructs the actor how to move in order to express the feelings of the role. Also, it is a very important factor how the actors look like from the audience. Therefore, actors often check their movements with a mirror.

- Feelings

Feelings in performances are expressions and perceptions that actors feel. An actor must speak lines with getting into the character. When the remark of actors is not enough of feelings, it is given a worthless performance. Therefore, directors coach for actors about feelings frequently. The grounds of remark of actors are feelings that the character has.

2.2 Methods of Recording Practice

Many theatrical organizations lack of places to practice and opportunities, therefore they encourage actors' self-practicing to improve the quality of performance. In self-practice, actors mainly review the previous lesson and prepare for the role by reading the script for the next lesson. Therefore, it is important to keep a record of these elements to improve the quality of the performance. The following methods can be considered for recording performance.

- Writing to the script by text

Recording is done by handwriting. Currently, this is the most common recording method. This method is difficult to record nonverbal elements such as body movement. In addition, there is a disadvantage that an actor must temporarily stop the performance to record.

- Movie

In this method, it is possible to record a lot of information and record nonverbal elements. Indeed, there are many actors to use for review by taking movies of their own acting. However, the recorded movie has no use other than watching it. In addition, there is a disadvantage that confirmation becomes complicated in proportion to the recording time. For example, if you record a two hour lesson by a video, it is hard work to find a few seconds of acting that you would like to check.

- Motion capture

This method is suitable for recording non-verbal elements such as body movements. Also, recorded data can be applied to various things. For example, it is possible to visualize a recorded motion data and confirm the performance from an arbitrary viewpoint. In addition, if you record the performance separately for each actor, it is possible to easily search for performance that the actor wants to confirm. On the other hand, there is a disadvantage that cost of motion capture equipment is high, and in some cases, additional time is required for motion capture.

Considering the use of the record of physical expression for practice, we think that high versatility of recorded data is important. According to research [5] by Ando et al., to improve actors' quality of performance, it is important for actors to review performance from an objective point of view. Take motion capture as an example, you can check the movement of your body from any viewpoint based on the data obtained by motion capture. Furthermore, it is possible to apply motion to avatar using that data, and various application methods using motion capture are conceivable. On the other hand, in the case of a movie, since the direction is fixed, it is impossible to confirm the viewpoint from the audience seats and opponent actors. Therefore, we focused on motion capture in this paper. In the next chapter, we introduce Digital-Script that handles elements necessary for theater as digital data.

3 DIGITAL-SCRIPT

3.1 Overview of Digital-Script

We propose Digital-Script which is added with the data necessary to performances. Digital-Script is handled as a database, and is different from the conventional scripts.

Theatrical production has technical elements and actor's performance elements. Technical factors are such as sound effect, lighting effect, stage carpenter and stage design. In actor's performance, “Timing of each actions”, “Body Movements”, and “Feelings” are very important elements. However, it is difficult to read these factors from a script written only in text. Therefore, in order to visualize these important elements, we try to handle them as digital data. Especially, in this paper, we focus on adding motion capture data of actor's body movements to Digital-Script. Upon making database of scripts, we compose Digital-Script by text and numerical value in order to make the system which has a variety methods of the visualization to each needs, such as voluntary exercise of actors, directions input of a director, and stage confirmation. In addition, in order to help actors grasp the timing of each action that changes on the stage, we add time information to each data. These data are added in the input applications of the digital script of the previous research. In this paper, we add motion data to Digital-Script.

3.2 Digital-Script Database

Digital-Script is handled as database which is implemented by MySQL. Figure 1 shows the configuration of the Digital-Script database. Digital-Script is distributed by scene unit. Scene data manages a certain scene and includes “SceneID”, "Title", "SceneName", and Actor data. In the following, we explain Actor data in detail. In Actor data, the peculiar ID and name are assigned to every character. In addition, "SAY", "FEELING", "MOVE", "SEE", and "ACTION" are included in it.

- SAY

SAY data are about actors' speech. This means that an actor speaks A when the elapsed time gets t sec. Strength means that an actor should strongly/weakly read an appointed part of lines A. Long means that an actor say lines A in l second.

- FEELING

FEELING data are about the feelings of an actor. This means that an actor has feelings E at the time of t second.

- MOVE

MOVE data are about actor's standing position. This means actor's movement to (x, y, z) coordinates during elapsed time between t1 sec to t2 second.

- SEE

SEE data are about actor's head direction where the actor should look at. This means that the actor looks at the designated direction when the elapsed time get t sec. The direction where the actor should look at is designated by (x, y, z) coordinates.

- ACTION

ACTION data are about actors' body movements. This means that the actor performances as M when the elapsed time get t second.

The above data can be used in various applications. Section 4.3 describes examples in which these Digital-Script data are read into the system and visualized. In this paper, ACTION is newly defined in Digital-Script and motion capture data is added to it.

4 RELATED WORK

In this section, we introduce several studies to support theatrical activities.

4.1 Recording of Body Movements

Hachimura et al. tried to motion capture the behavior of Japanese dance, which is a traditional Japanese art, to make it useful for inheritance and preservation [6]. In addition, Hachimura and colleagues analyze the characteristics of "Okuri" which is the basic behavior of dancing by using motion-captured data. There are also many studies using motion capture in the field of ballet and dance. Saga and colleagues have studied using motion capture for simulation of ballet choreography [7]. Their study's goal is to develop an automatic composing system for ballet by using motion Data archives. However, these studies use large-scale equipment for motion capture, and Hachimura also states that the operation of these equipments requires experts with special skills.

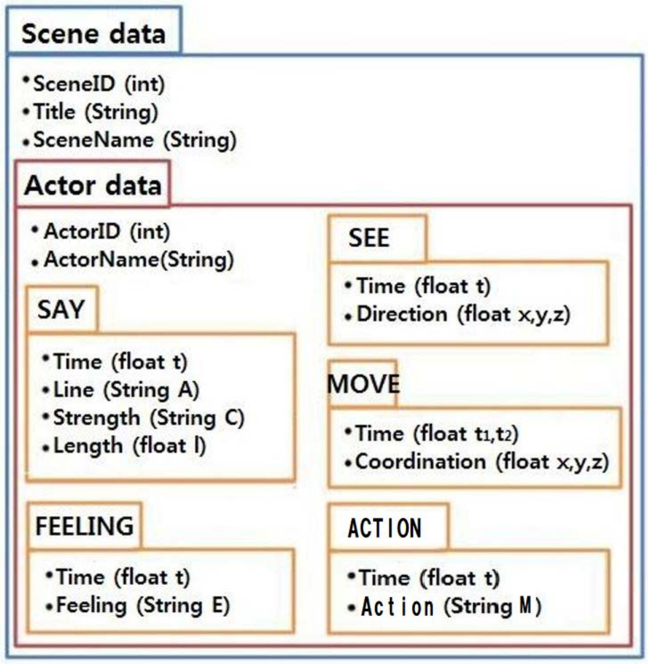


Figure1: Components of Digital-Script Database

4.2 Action Cueing

Takatsu et al. applied smart watch to support the practice of dramas [8]. In order to practice smoothly, it is important for the actors to learn the correct order of performance. However, in the early steps of practice, it is difficult for actors because a play script has a lot of information to learn. In this research, Smart watch has the role of telling the information to the actors and informing them in the correct order. Smart watch is a way to convey information without putting a physical burden on the actor. This research makes it easier for actors to perform in the proper order, and actors can concentrate more on acting.

4.3 Digital-Script

We introduce two studies to support practice of theater by using Digital-Script.

Shimada et al. proposes the self-practice system [9] which reflected a direction of a director. This focuses on timing of an action such as beginning of lines and movements. This system visualizes Digital-Script in two ways like Fig. 2. In this display, timing of each action is showed by a time-line and spatial information is showed by 3DCG. By using this system, it is possible for an actor to practice while grasping progress of a performance intuitively.

Theater organizations may not practice anytime with all members because they don't have their own practice space and most actors have another job. By using Digital-Script, theatrical practice support system [10] enables actors to practice performance in the situation that a director or a part of actors cannot participate. This system needs monocular Head Mounted Display (HMD). In this HMD, virtual actor that plays absent actor's role is displayed, and a user is coached automatically by detecting and comparing actor's movement with information in Digital-Script like Fig. 3.

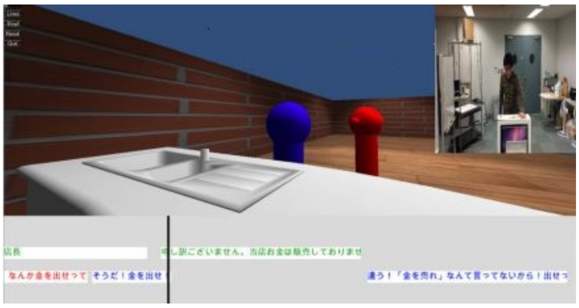


Figure 2: Voluntary practice system [9]

Figure 3: Theatrical practice support system [10]

5 AUTOMATIC MOTION CAPTURE SYSTEM OF ACTORS' BODY MOVEMENTS

Recording performance is important in the practice of theaters. However, with the current method, in practice of acting, recording body movements is sometimes hard. Therefore, we propose the system that can automatically record actors' body movements while supporting for actors' learning. This system can notify action cueing to actors during recording body movements. This automatically generates motion data without long time and heavy workload. By notifying action cue to actors, it can make practice smooth and reduce mistakes of acting. The contributions of this system are as followings.

- Simplification of recording body movement

By automation of the recording of the actors' physical expressions, there is no need to perform additional work for recording. This system does not require special skills or more time than practice time. This contribution can solve the problem that the recording operation shortens the practice time.

- Data acquisition of body movement as Digital-Script

Scripts are what represent story of the theater. However, since scripts are written in text, actors cannot intuitively grasp the non-verbal elements such as the movement of the body and the standing positions of the actors. Therefore, we propose the motion capture system for the physical expressions of actors. We treat this recorded motion data as a part of Digital-Script and save it in a format that can be used for various purposes.

- Notifying of the order of performance of actors

In this system, smart watch notifies the order of performance of each actor individually. Smart watch shows what the actor will do next and the next line. Smart watch's vibration informs the order of acting, so the actor does not need to see the screen. By using this system in the early stage of practice, the actor can learn the flow of acting by notifying action order. This system can automatically record the body movement while actors concentrate on performance.

5.1 System Overview

This system automatically records the performance without a heavy burden for an actor or a director. Also, this system notifies actors of the order of acting and supports the practice of actors. Figure 4 shows the usage of this system. This system requires PC, Kinect, tablets, and smart watches. The director uses the web application on the PC. The Fig. 5 shows the screen of the web application. On the right side of this screen, the result of reference to Digital-Script database is displayed. Lines, timings, actor's names, actions are shown. The director checks the performance of the actors while confirming the flow of the script on this screen. By pressing the start button, practice begins along the data of Digital-Script. On the left side of this screen, the director can select the line and resume practice from there. Also, the director can delete or add lines on this application. All the results of editing are automatically saved in Digital-Script.

When the performance starts, this system tracks progress of performance automatically according to Digital-Script, and notification is sent to the actor's smart watch. At that time, all movements of actors are motion captured by Kinect and saved as Digital-Script. On the other hand, the actor wears a smart watch and practices. The smart watch notifies by vibrating according to the behavior of each actor and the timing of speech. This timing of notification is based on the Digital-Script database.

Figure 6 shows the notification display of smart watch. On the display of smart watch, the line that the actor speaks at that timing is displayed. Therefore, actors can reduce mistakes in acting and can practice smoothly. This function plays the role of "prompter" in an actual theater that gives a reminder to actors who forgot the line etc.

We use Samsung Gear Live for smart watch. Samsung Gear Live is designed to be used in paired with an Android tablet, and an Android tablet is required for communication with the server of this system. Figure 7 shows the display of Nexsus 7, and this is the interface that cooperates with PC and smart watches. You can grasp the status of the connection with PC and smart watches, and the status of this system on this screen.

Figure 4: Usage image of this system

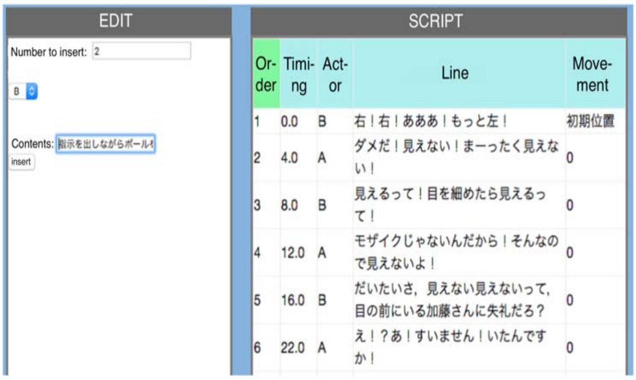


Figure 5: Web application

Figure 6: Notification window of smart watch

Figure 7: Tablet display

5.2 Use Case

We assume that this system is used in the following cases.

- Early stages of practice

This system is supposed to be used at the stage of practice. In early stage of practice, actors do not remember the order of acting perfectly. Normally, at this stage, the actors practice while having a script, and receive instruction from the director.

- 4 meters square space

This is the extent of space where actors can practice while moving. This range of space depends on specifications of Kinect.

- Participating of the director in practice

We assume that the director participates in the practice and instructs the actors' acting. The director has the right to decide all performance. Therefore, we assume that theatrical organizations use this system with the aim of recording of performance.

5.3 Implementation

Figure 8 shows the hardware architecture of this system. This system requires PC, Kinect, tablets, smart watches. We used Samsung Gear Live for smart watches and Nexsus 7 for Tablets. Samsung Gear Live is connected to the PC server by WebSocket. In order to connect quickly and support multiple devices, we use WebSocket. WebSocket can keep socket open with bidirectional connection and support stable connection in multi device environment. This was implemented by node.js and was connected to the device using Cloud9 and heroku. In this system, each actor wears this smart watch and receives action cueing by vibration. Samsung Gear Live is paired with a Nexus 7 tablet. Since only one smart watch can be controlled with one tablet, we used multiple tablets connected by Bluetooth. We use Android Wear 5.0.1 as the implementation platform and the Android Java API v21.1.2. The processor is 1.2 GHz Qualcomm Snapdragon 400, and the display is Super AMOLED 1.63 type. Tablet OS is Android Wear 4.4.3, 1.3 GHz NVIDIA Tegra 3 mobile processor. Kinect for Windows API v2.0 was used as a depth camera to recognize the movement of each actor. Kinect has a depth camera, RGB camera, multiple eye ray microphone, and processor. Kinect can recognize the gesture, the position, and height of the users. Figure 9 shows summary of the network configuration of this system.

5.4 Motion Capture

In a theatrical practice, multiple actors participate it. Since this system can detect up to six actors, it is possible to record movements of multiple actors at the same time. This system always detects actors during practice. Then, motion capture is started automatically when action cue is notified to the actor. After the start of motion capture, skeleton coord inates of the actor are recorded for each joint for few seconds. The recorded data is output as a CSV file and finally converted to a Digital-Script format. Figure 10 shows an example of visualized Digital-Script data.

6 EVALUATION

This chapter describes experiment to evaluate workload of this system in recording actor’s body movements.

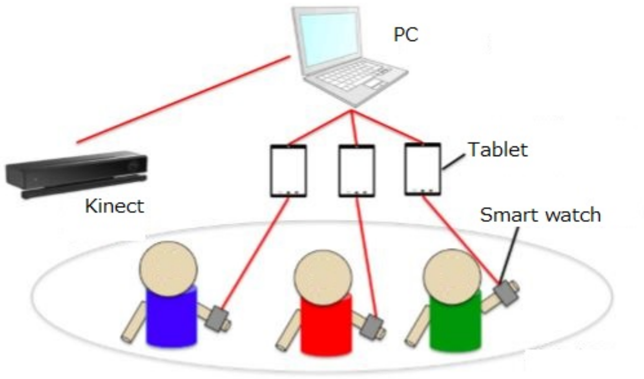


Figure 8: Hardware architecture

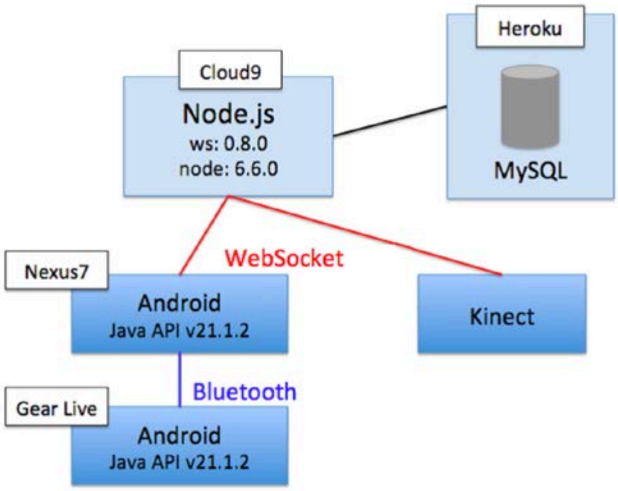


Figure 9: Network configuration

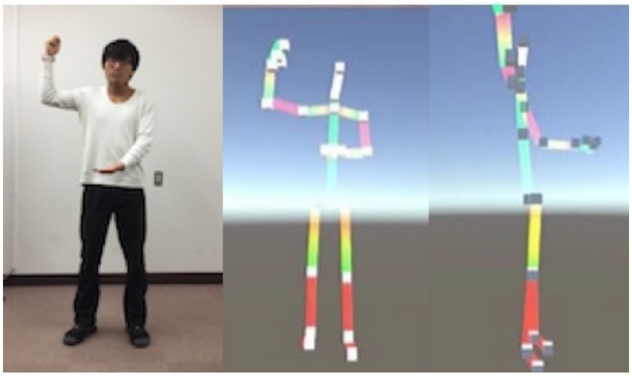


Figure 10: Visualized image

6.1 Subjects

10 university and graduate school students participated in this experiment. One subject participated in one experiment.

6.2 Method

As a director, the subject instructed the actors about performances while watching the script. Two experimenters participated in the experiments as the actors. Subjects recorded performances of actors. We conducted the experiment in two situations: Experiment A and B. In Experiment A, subjects recorded body movements of actors with this system, and in Experiment B, subjects record it with human-shaped device. Five of the ten subjects belonged to Group 1. Firstly, Group 1 participated in Experiment A and then did Experiment B. The remaining 5 subjects belonged to group 2. Group 2 first participated in Experiment B and then did Experiment A.

In Experiment A, the subject participated in this Experiment As a director. First, the subject saw the performance of the actor. Subsequently, the subject instructed the acting of the actors. The content of the instruction was what the experimenters informed the subject in advance. After those instructions, the actors performed acting according to instructions. The subject recorded their performance using this system. We measured the time required to record performance. To summarize, Experiment A proceeded in the following procedure.

1. Explanation of the flow of the Experiment and the operation of this system
2. Watching practice
3. Instructing to the actors
4. Recording practice

In Experiment B, like Experiment A, the subject participated in this experiment as a director. The subsequent flow is the same as Experiment A, but in this experiment this system was not used. First, the subject saw the performance of the actor. Subsequently, the subject instructed the acting of the actors. After those instructions, the actors performed acting according to instructions. We recorded that situation in video. The subject created motion data using QUMARION (see Fig. 11) while referring the video. QUMARION is a human type input device developed and sold by Celsys [11], Inc. It is a device used for creating motion and pose of 3D animation. The size of this device is 29 × 30.8 × 10.8 (cm) and the weight is 255g. It has 16 joints and 32 sensors and can create motion intuitively and easily. We measured the time required by subjects to make motion using this device. To summarize, Experiment B proceeded in the following procedure.

1. Explanation of the flow of the experiment and the operation of this system
2. Watching practice
3. Instructing to the actors
4. Practice with recording by video
5. Creating motion data by QUMARION with reference to movie

We used a script for each experiment. We used two scripts, X and Y, from free script download service [12]. Details of the two scripts are shown in the table 1. Both scripts are for a theater of around 150 seconds. For both scripts, we prepared six experimental tasks.

6.3 Evaluation Items

In both experiments, we evaluated the workload and required time in the recording of the physical expression of the actors.

Figure 11: QUMARION [11]

Table 1: Details of scripts

	Length	Number of lines	Number of tasks
Script X	150 sec	30	6
Script Y	150 sec	32	6

To evaluate workload of with this system, we used NASA - Task Load Index (NASA-TLX, here in after referred to as TLX). TLX is one of the most frequently used subjective workload metrics. TLX has 6 scales: Mental Demand, Physical Demand, Temporal Demand, Overall Performance Frustration Level. They are rated for each task within a 100-points range with 5-point steps. Figure 12 shows part of the answer sheet for these scales. Subjects fill in this line the degree of involvement in the workload for each scale. We measure the position of the mark on the line to get a rating of 1 to 100 ($=\nu_i$). We create an individual weighting of these scales by letting the subjects compare them pair-wise based on their perceived importance. This needs subjects to choose which scale is more relevant to workload. The number of times each is chosen is the weighted score ($=\omega_i$). We calculated WWL (weighted workload) from this weighted score by formula (1). A small WWL value means that the participant's workload is small.

$$WWL = \frac{\sum_{i=1}^{6}(\omega_i \times \nu_i)}{\sum_{i=1}^{6}\omega_i} \qquad (1)$$

In Experiment A, we measured the time taken for recording of the physical expressions of the actors. On the other hand, in Experiment B, we measured the time taken to create motion data by QUMARION.

6.4 Results and Discussion

Table 2 shows group 1's result about 6 scales of NASA-TLX, and table 3 shows group 2's. Table 4 shows the results

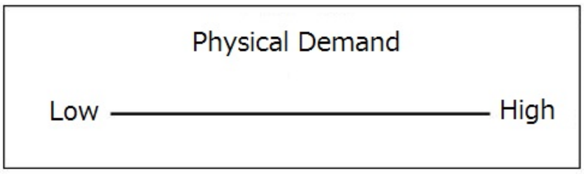


Figure 12: Workload metrics

Table 2: Result of NASA-TLX for group 1

Scale	Proposal method	Compared method
	Score	
Physical demand	27.4	49.4
Mental demand	36.6	43.6
Temporal demand	40.2	18.2
Performance	33.8	57.6
Effort	16.5	62.8
Frustration	27.2	52.2
WWL (Weighted workload)	33.8	56.5

Table 3: Result of NASA-TLX for group2

Scale	Proposal method	Compared method
	Score	
Physical demand	33.2	59.6
Mental demand	37.8	59.8
Temporal demand	15.8	18.2
Performance	84.3	47.2
Effort	36.2	38.2
Frustration	34.0	43.8
WWL (Weighted workload)	38.3	51.1

Table 4: Workload of group 1

Experiment	Script	WWL(Weighted workload)
A	X	33.8
B	Y	56.5

Table 5: Workload of group 2

Experiment	Script	WWL(Weighted workload)
A	X	38.3
B	Y	51.1

Table 6: Results of time required

Experiment	Group 1	Group 2	Ave.
A	223.4 sec	195.2 sec	209.3 sec
B	827.2 sec	626.0 sec	726.6 sec

for group 1's workload, and Table 5 shows group 2's. Table 6 summarizes the results on the time required for each group of experiments.

From table 4 and 5, it can be seen that the values of WWL are smaller in Experiment A in both groups. From this, we found that this system can record actors' body movements with low workload.

From table 6, we see that the time taken by Experiment A is shorter. From this result, it can be said that subjects record body movements with this system by the shorter time. We assume that one of the reasons for reducing workload is shorter time required.

From table 1 and 6, we found that Experiment A takes about 209 seconds compared with the time of the scripts of

150 seconds. The time required by this system is slightly longer than the script time, but we think that recording can be completed in a sufficiently short time.

7 CONCLUSION

Self-practice of actors is encouraged by many theater organizations. In a self-practice, an actor reviews important elements in performance such as body movements. Therefore, during practice, to record instructions about it from the director is very important to blush up actor's performance. Currently, the recording is done mainly by writing text in a script. However, this method is not suitable for recording actors' body movements and it is hard for an actor to review during self-practice. Therefore, we tried using motion capture to deal with this problem. Generally, motion capture requires long time and workload. In this paper, we propose the system that simplifying records body movements during practice by motion capture. In addition, by notifying the actor of the action cue using the smart watch, this system reduces mistakes in performance of actors and makes practice smoother. We conducted experiments to evaluate the workload of this system and time required. To evaluate workload of with this system, we used NASA-TLX. TLX is one of the most frequently used subjective workload metrics. As a result, we confirmed that the system can record in short time with small workload.

REFERENCES

[1] Agency for Cultural Affairs, Survey on training and utilization of human resources related to demonstration artists. (in Japanese)
http://www.bunka.go.jp/seisaku/bunkashingikai/seisaku/07/03/pdf/shiryo_3.pdf
[2] Tsumura, T., Tsubochi, E., Theater Creation Manual, Japan Foundation for Regional Art-Activities, (2006).
[3] Goan, M., "System dynamics on the creation of drama-making processes," Jpn. Cogn. Sci. Soc. Cogn. Sci. 14(4), pp. 509–531 (2007). (in Japanese)
[4] Kameoka, Y., Engeki yarouyo! , Seikyusha, pp. 168-170, (2004). (in Japanese)
[5] Ando, H., "Expertise of Actors in Assuming the Viewpoint of the Character, the Actor, and the Audience, "Kyoto University Research Information Repository 52, pp. 386-398, (2006). (in Japanese)
[6] K. Hachimura, "Digital Archiving of Dancing by Using Motion Capture, Computer Vision and Image Media," pp. 1-8, (2007). (in Japanese)
[7] A. Soga, B. Umino, "Creative Simulation by Analyzing Choreographic Elements of Classical Ballet, IPSJ SIG Computers and the Humanities," pp. 45-52, (2008). (in Japanese)
[8] R. Takatsu, N. Katayama, H. Shigeno, and K. Okada. "A Wearable Action Cueing System for Theatrical Performance Practice," Collaboration Technologies and Social Computing: 8th International Conference, CollabTech 2016, Proceedings. Vol. 647. Springer, (2016).
[9] Shimada, M., Fujishige, S., Okada, K.-I., "Supporting actor's voluntary training considering the direction of drama." IPSJ Trans. Digit. Content 4(1), pp. 1–9 (2016). (in Japanese)
[10] Shimada, M., Takayoshi T., Hiroshi, S., Okada, K.-I., "Supporting Theatrical Performance Practice by Collaborating Real and Virtual Space." Collaboration Technologies and Social Computing: 8th International Conference, CollabTech 2016, Proceedings. Vol. 647. Springer, (2016).
[11] CELSYS, Inc.
https://www.celsys.co.jp/
[12] Free script download service: Hariko no tora no ana.
https://haritora.net/script.cgi

(Received October 20, 2017)
(Revised December 26, 2017)

Takayoshi Takano
B.S. degree from the Department of Information and Computer Science of Keio University, Japan in 2016. He is currently working toward an M.S. degree in Open and Environment Systems at Keio University. His research interests include computer supported cooperative work.

Hiroshi Shigeno received his B.S., M.E. and Ph.D. degrees in instrumentation engineering from Keio University, Japan in 1990, 1992, 1997. Since 1998, he has been with the Department of Information and Computer Science at Keio University, where he is currently a professor. His current research interests include mobile and ubiquitous computing, network architecture, and Intelligent Transport Systems. He is a member of IEEE, ACM, and IEICE.

Ken-ichi Okada received his B.S., M.E. and Ph.D. degrees in instrumentation engineering from Keio University, Japan in 1973, 1975, 1982.Since 1998, he has been with the Department of Information and Computer Science at Keio University, where he is currently a professor emeritus. His research interests include CSCW, groupware, human computer interaction, and ubiquitous computing. He has published140 journal papers, 170 international conference papers, and 18 books entitled "Collaboration and Communication", "Designing Communication and Collaboration Support Systems", "Introduction to Groupware" and so on. He is a member of IEEE, ACM, IEICE and IPSJ. He was a chair of SIGGW, a chief editor of IPSJ Magazine, and chief editor of IPSJ Journal. Dr. Okada received the IPSJ Best Paper Award in 1995, 2000, and 2008, the IPSJ 40 the Anniversary Paper Award in 2000, IPSJ Fellow in 2002 and the IEEE SAINT Best Paper Award in 2004.

Regular Paper

A Distributed Internet Live Broadcasting System Enhanced by Cloud Computing Services

Satoru Matsumoto[*, †], Yoshimasa Ishi[*], Tomoki Yoshihisa[*], Tomoya Kawakami[*, **], Yuuichi Teranishi[*, ***]

[*]Cybermediacenter, Osaka University, Japan
[**]Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[***]National Institute of Information and Communications Technology, Japan
[†]smtsumoto@cmc.osaka-u.ac.jp

Abstract -Recently, live Internet broadcasting services such as USTREAM and TwitCasting have become popular. In such broadcasting, broadcasters record videos using the cameras on their laptops or smart phones (clients). The clients send the video data to the broadcast servers of live Internet broadcasting services, and the servers distribute the videos to the viewers. We can use high processing power computers provided by cloud computing services to improve the qualities of the videos or to show additional information. The processing power of the computers provided by cloud computing services often changes. Thus, by flexibly selecting the most useful computers, and flexibly determining the processing from the desired effects, clients can reduce the limitations on the types of effects. In this paper, we propose a 'different-world broadcasting system', a live Internet broadcasting system, enhanced by cloud computing services. In the proposed system, the policy for using cloud computing services, (i.e., the selection of the desired machine and effect specifications,) is described by ECA (Event, Condition, Action) rules, to enable flexible policy change. In this research, we implemented and evaluated the different-world broadcasting system enhanced by cloud computing services.

Keywords: Streaming Delivery, Multimedia, Internet Live Broadcasting, Video on Demand, Different Worlds Broadcasting

1 INTRODUCTION

In recent years, real-time video distribution (live Internet broadcasting) via the Internet, such as USTREAM and TwitCasting, has become popular. In live Internet broadcasting, broadcasters or viewers often add video effects to enhance visual experiences. By using high processing power clients, the broadcasters can add various types of effects to further enhance the visual experiences and/or show additional information. Meanwhile, high processing power computers, provided by cloud computing services such as Microsoft Azure, Amazon Web Services, and Google Cloud Platforms, have recently become available. For using these high power computers to add such effects, clients select the desired computer, send their recorded video data to the selected computer, and specify the desired effects (types, parameters, etc.)

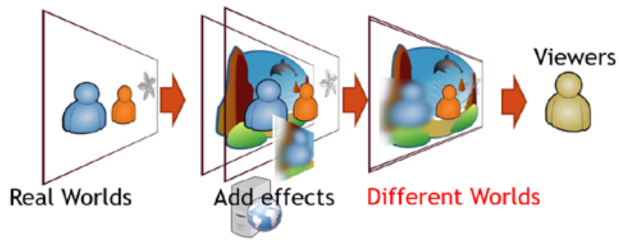


Figure 1: Different worlds broadcasting

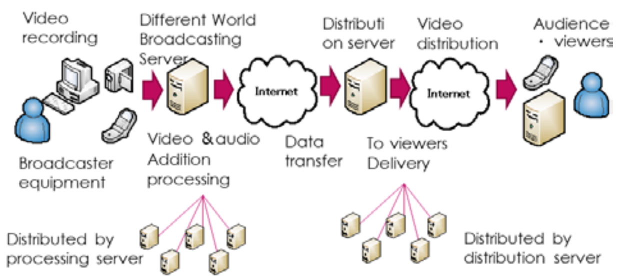


Figure 2: Overview of load distribution of effect addition processing

Though some distributed processing systems for live Internet broadcasting have been developed [1],[2], they do not focus on cloud computing services. Our research group has developed a distributed live Internet broadcasting system, called the different-world broadcasting system [1], an image of the system is shown in Fig. 1. In different-world broadcasting, effects such as blurring are added by a camera to videos recording the real-world, and videos are distributed to viewers. In the different-world broadcasting system, the different-world broadcasting server adds the video effect. Therefore, the processing load caused by the addition of the video effect, which has conventionally been done by the clients, is here distributed (Fig. 2). This system assumes that a different-world broadcasting server is a computer owned by a distributor, or a virtual computer provided by a cloud.

In our previous research [1], we developed a different-world broadcasting system using a computer owned by a distributor as the different-world broadcasting server. In this case, when many viewers request adding effects in a short period of time, and when effects with a large load are requested, it takes long time to add the effects since the computer processes all of these requests. This causes delay and destabilization of the delivery frame rate. For this reason, the available effects in this system are limited, to light-load effects, in order to keep the delivery frame rate. This limitation decreases the quality of the user's different-world experience.

In this research, we assume the systems that use a cloud service virtual computer as the different-world broadcasting server that adds video effects. In general, a number of virtual machines are easily available in cloud services. Therefore, the use of multiple virtual machines as different-world broadcasting servers improve the speed to add effects, while distributing the load among different-world broadcasting servers and maintaining a stable delivery frame rate. In our previous research, we assumed the systems that use only a single computer. In this research, we implement a distributed live Internet broadcasting system using cloud service, and evaluate its performance. In the implemented system, effect addition is processed using the virtual computer provided by cloud services. When delegating processing to virtual computers, the performance of the processing computer is flexibly controlled by ECA rules. The evaluation results confirmed that the turnaround time of the effect addition can be reduced by using these rules.

The rest of the paper is organized as follows. We discuss related research in Section 2. In Section 3, we explain the design and the implementation of the proposed different-world broadcasting system using cloud service, and in Section 4, we evaluate its performance. We discuss the evaluation in Section 5, and conclude the research in Section 6.

2 RELATED RESEARCH

Various methods have been proposed for distributing the processing load. However, most of them require the construction of load distribution systems beforehand. In live Internet broadcasting, however, it is easy to commence live broadcasting, and the number of video distribution terminals is typically unspecified. Therefore, it is difficult to construct a load distribution system for a large number of video distribution terminals in advance. Several studies have been done on distributed video processing systems that use computers other than video recording terminals.

In MediaPaaS, designed as a cloud computing service in the PaaS (Platform as a Service) type for live broadcasting, it is possible to encode, re-encode, and distribute the video using the computer on the service provider side [2]. Different from MediaPaas, our proposed system implements, a load balancing system by using PIAX [9], a P2P agent platform.

In our proposed system, when the video recording terminal connects to a different-world broadcasting server, it interrupts the connection and distributes the load by randomly selecting the connection destination from some of different-world broadcasting servers. It has been experimentally confirmed that distributing the processing load over a number of different-world broadcasting servers and video recording terminals reduces video processing time. In this research, in order to be able to use more different-world broadcasting servers, we implement the proposed system, and evaluate the video processing time, using the virtual computers provided by a commercial cloud service.

The system implemented in [3] enables the viewing of recently recorded video (chasing playback) for live broadcasting performed by a P2P network. In this video distribution system proposed in this paper, broadcasters select adding video effects, one of the features of the system proposed in this research.

Several methods have been proposed for reducing the delay time of video distribution for live Internet broadcasting. In live broadcasting with SmoothCache 2.0 [4], it is possible to reduce the communication load and delay time of the video recording terminal, by caching video data from other peers and distributing it from the cached peers using the P2P network. Dai et al. Proposes a P2P network distribution route determination method which minimizes the delay time in [5]. In the HD method proposed, communication is reduced by simultaneously transmitting video data to multiple viewing terminals using one-to-many broadcast distribution combined with one-to-one communication in [6]. In the proposed system, these delay reduction methods can be applied when distributing videos, but our research differs in its design of the video processing system.

studies on systems that perform video processing on stored video data have been discussed. Gibbon et al. proposed a system that performs video processing by transferring video data shot by a camera to a computer with high processing power. Ting et al. proposed a system that stores images captured by low processing power computers (e.g., smartphones) directly in an external storage device such as cloud storage, without storing them in them [8]. These systems, however, target stored video data, and cannot be applied for live Internet broadcasting.

3 SYSTEM DESIGN AND IMPLEMENTATION

We applied a load distribution mechanism to the different-world broadcasting servers and video recording terminals of the different-world broadcasting system.

In this section, we explain the system design, the load distribution mechanism, the design of the description method used for dynamically allocating and executing the effect addition processing, and the implementation of these components.

3.1 Design Overview

In the different-world broadcasting system enhanced by cloud services in order to distribute the load of virtual machines, it is necessary to describe the processing to be assigned to each virtual machine. The types of processing can be explicitly described and allocated when the types do not change frequently. When they frequently change, the system dynamically allocates them according to the specific type of processing. In this research, processing is allocated using ECA rules, so that the allocations can be flexibly inscribed.

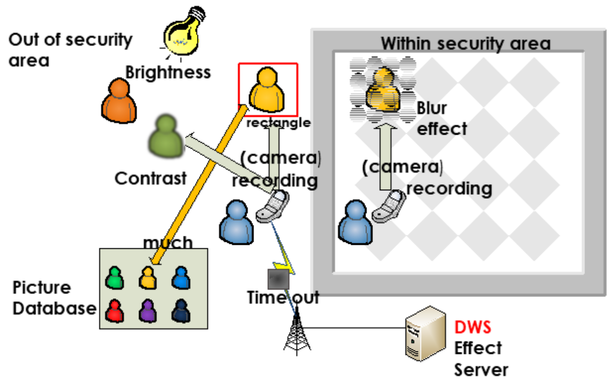


Figure 3: An image of video processes on different-world broadcasting systems

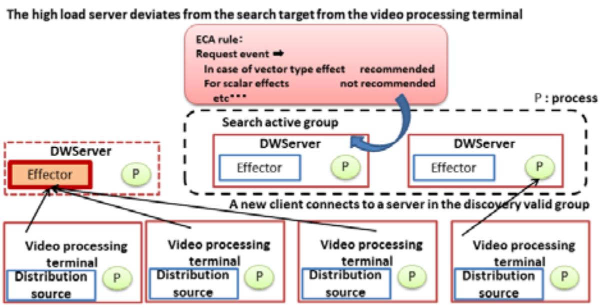


Figure 4: Overview of ECA rules application

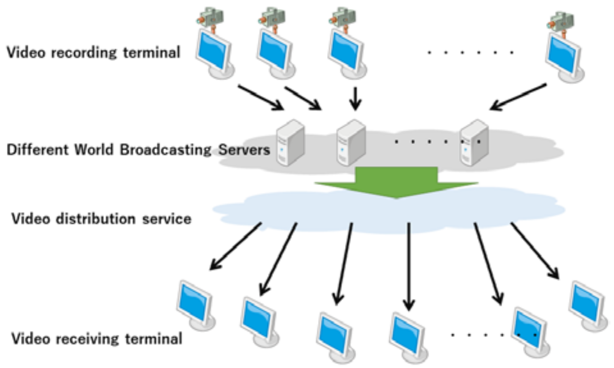


Figure 5: An image of video distribution

In our previous system [1], in order to reduce the operation load on broadcasters, event driven processing based on ECA rules is automatically executed. At the time of recording, each rule is automatically invoked by its specified event, and enlists different-world broadcasting servers to add visual and auditory effects. For example, to protect portrait rights, face authentication automatically detects specific persons in a database and adds mosaics effects on these faces when the individuals enter a given area. Also, the system can automatically adjust the brightness of the video or adjust the contrast according to the brightness of the image (Fig. 3). In our previous research, ECA rules were applied on the different-world broadcasting server side, mainly for the purpose of reducing the operation load on the broadcaster, as described above. In this research, we design a load balancing rule based on ECA rules, so as to avoid new video processing requests to different-world broadcasting servers in a heavy load state. The ECA rules are distributed from the ECA rule distribution server to each server (Fig. 4).

3.2 Overview of Load Balancing Mechanism

Figure 5 shows the environment assumed in this research. In our assumed environments, there are three types of terminals or servers: video recording terminals, different-world broadcasting servers, and video receiving terminals. The video recording terminal selects a different-world broadcasting server that can execute desired video effects, and transmits video effect libraries and the recorded videos. The different-world broadcasting server operated on the cloud service virtual machine executes the video processing to the video sent from the video recording terminal, according to the requests from the terminal. The video processed by the different-world broadcasting server is delivered to the video receiving terminals via the video distribution service. The video receiving terminal receives the processed video after selecting the server or the channel of the video distribution service.

In this research, focusing on how the video recording terminal selects a different-world broadcasting server, we consider load distribution for different-world broadcasting servers. In our previously proposed system, load distribution is achieved by connecting the video recording terminal via the load distribution mechanism, when connecting to the different-world broadcasting server. In this method, the load distribution mechanism selects a different-world broadcasting server that less video recording terminals use when it is necessary to switch to another different-world broadcasting server while the video is being transmitted. For example when switching to a different-world broadcasting server more suitable for a given scene or effect, the video recording terminal terminates the existing connection, which cannot be resumed without formal reconnection. For this reason, it is difficult to smoothly switch between different-world broadcasting servers while continuing the video distribution.

In addition, even when the load distribution mechanism changes the different-world broadcasting server according to the requests from video recording terminals, the mechanism relays the communication data as a simple byte stream. This means that the data structure (e.g., the boundary of the video frame) is ignored, and the video data may be damaged when the mechanism switches the servers. Therefore, in this research, the load distribution mechanism serves only to select the different-world broadcasting server, based on the requests of video recording terminals, and the video recording terminals connects directly to one of the different-world broadcasting servers as a result of these requests. In this way,

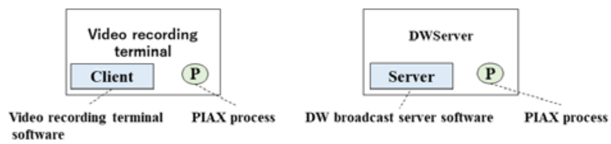


Figure 6: Internal systems of video recording terminals and different-world broadcasting servers

video recording terminals can switch different-world broadcasting servers at any time, even when the boundaries of the scene or video frames come.

3.3 System Architecture

Figure 6 shows the internal configurations of the video recording terminal and different-world broadcasting server. Video recording terminal software and a client-side PIAX system are installed to the video recording terminal. Different-world broadcasting server software and a server-side PIAX system are installed to the different-world broadcasting server. PIAX [9] is a Java-based platform middleware that enables serverless and efficient resource searches by utilizing the search function of the overlay network. PIAX is provided as open source software. The PIAX system in the video recording terminal and the different-world broadcasting server can be connected via their overlay network.

The client-side PIAX system on the video recording terminal searches the overlay network in accordance with the request of the video recording terminal software. The system selects a different-world broadcasting server from a host of different-world broadcasting servers, and returns the IP address of the selected server and the listen port number of the server software. Based on the response, the video recording terminal software establishes a connection with the different-world broadcasting server software and starts sending the video. JSON-RPC is used directly over TCP for communication between the video recording terminal software and client-side PIAX system.

The server-side PIAX system on the different-world broadcasting server determines its search state based on the instructions from the different-world broadcasting server software. If the different-world broadcasting server is regarded as in a high-load state itself, it connects with the control port of the server-side PIAX system on localhost, and sends a ‘leave’ request command for the search, which prevents the different-world broadcasting server from receiving further connection from the video recording terminal. If the load on the different-world broadcasting server lessens, and the server gets a margin to accept a new connection, the broadcasting server reconnects with the control port of the server-side PIAX system on localhost, and sends a ‘release leave’ request command. This command, the server-side PIAX system again enters a state of search acceptance from the video recording terminal.

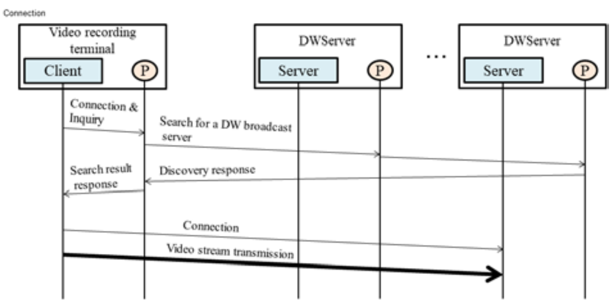


Figure 7: Data flow and timing chart of communication procedure

```
{
  "Rule A":{
        "eventname":"Set_effect",
        "condition":{
                    "name":"Num_Find_Object",
                    "object":"object1_haar.xml",
                    "Value":">=1"
        },
        "action":{
                "name":"REQ_IP",
                "IP_address":"192.168.0.5"
        }
  },
  "Rule B":{
        "eventname":"Set_effect",
        "condition":{
                   "name":"Blur",
                },
        "action":{
                "name":"REQ_IP",
                "IP_address":"192.168.0.6"
        }
  }
}
```

Figure 8: An example ECA rules

Figure 9: A screen shot of different-world broadcasting server software

With the above operation, overload of the different-world broadcasting server software can be avoided since the new connection from the video recording terminal is controlled based on the load state of the different-world broadcasting server. Figure 7 shows the data flow and timing chart of the communication procedure.

3.4 Designing ECA Rules

The lists of events, conditions, and actions are explained in our previous paper [1]. In addition to the list, we add one event and one action to enhance the system for using cloud computing services. “Set_effect” events occur when new video effects are used in the system. “Req_IP” actions initially set the IP address of the different-world broadcasting server that adds the video effect designated in the condition part. In case when the designated server is on heavy loads, the address is automatically changed by PIAX. Figure 8 shows a concrete ECA rules set. In the example, the video recording terminal initially requires the video processing named "Num_Find_Object" or "Bluer" to the server of that IP address is 192.168.0.5 or 6.

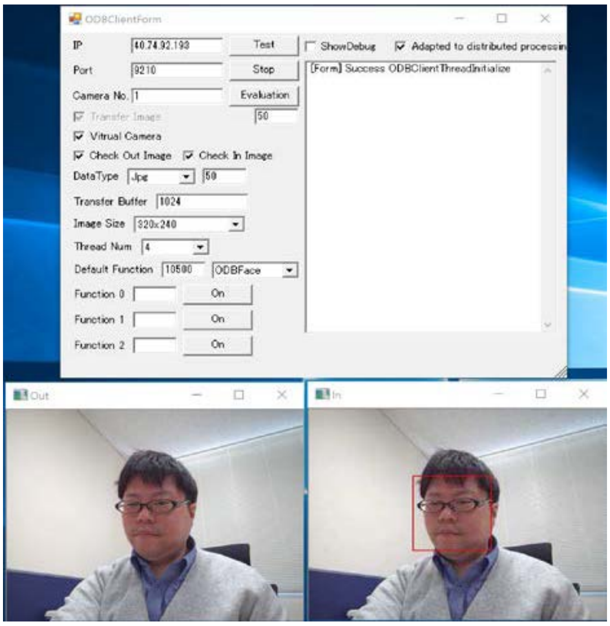


Figure 10: A screen shot of different-world broadcasting client software

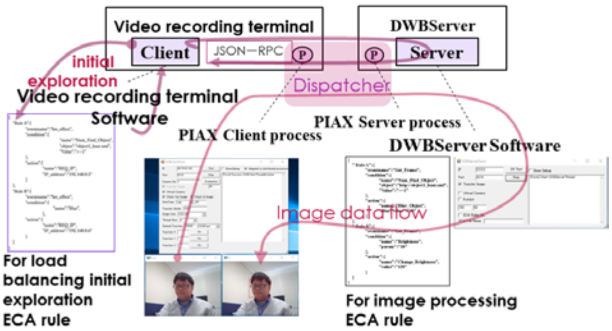


Figure 11: Processing flows of our implemented system

3.5 Implementation

We used Microsoft Azure as a cloud service, to deploy the different-world broadcasting system. The different-world broadcasting server works on virtual machines provided by the Azure service. Each virtual machine is logically connected by a Virtual Network (VNet), one of the services provided by Microsoft Azure. Figure 9 shows the user interface of the different-world broadcasting server software when initiating the video effect addition process. Figure 10 shows the user interface of the video recording terminal software that distributes the video on the video recording terminal. It can visually recognize the result of applying the selected effects.

If the 'ECA Rules On' check box in the different-world broadcasting server dialog box is checked, the image effect is automatically added, based on the different-world broadcasting server ECA rule without the operations of the user.

The IP address of the different-world broadcasting server for requesting the video processing is held in the video recording terminal software. The terminal software uses the IP address to request the video processing designated in the pull-down menu to the different-world broadcasting server if the 'Apply Distributed Processing' check box in the video recording terminal software dialog box is checked, the initial video processing request is sent to the different-world broadcasting server via the load distribution mechanism, based on the ECA rules held by the client software. The processing request is sent to a different-world broadcasting server with a low processing load. In this case, the ECA rule inscribes the initial IP address to request its execution to the load distribution mechanism. Figure 11 shows the processing flows of our implemented system in this research.

4 EVALUATION

We evaluate the effectiveness of the proposed method using an implementation system built on the virtual machines provided by the Microsoft Azure service. The evaluation method, environment, and results are described below.

4.1 Evaluation Method

To evaluate the efficiency of the proposed system, we focused on the video processing time and turnaround time, including the processing time of the ECA rule, as evaluation indices. We made a comparative evaluation of the turnaround time when (a) the video effect processing requests were concentrated on the video effect processing server without employing the ECA rule; and (b) the video effect processing requests were not concentrated, and the ECA rule was employed. For the evaluation, we assigned multiple computers with the same performance parameters on the cloud service. For the selection of available different-broadcasting servers, we used the PIAX overlay network explained in the previous section. When a different-world broadcasting server is overloaded, the server sent a notification to the server-side PIAX process, and wait until the load becomes low. We measured the turnaround time in two cases. one is the concentrated case, in which four video recording terminals made requests to one out of five different-world broadcasting servers. This corresponds to the situation (a). The other is the non-concentrated case, in which each of the four video recording terminals made requests to a different-world broadcasting server service. This corresponds to the situation (b).

Since the main focus of the evaluation is processing load distribution, live broadcasting software is not used in this evaluation. In the evaluation, face detection video processing, which detects the face of a person in a given video, is employed, with the video effect inscribed in the ECA rule. For evaluation under different computational loads, we gradually increased the load caused by human face detection and measured the turnaround time. The time taken to transmit and receive frame data was defined as the turnaround time. This includes the following four items.

Table Ⅰ: Specifications of virtual machine (different world broadcasting server)

Component	Performance
OS	Microsoft Windows Server 2016 Datacenter
Microsoft Azure (Virtual server service plan)	Standalone Server Microsoft Corporation Virtual Machine x64-based PC
CPU	Intel E5-2697 v3 Equivalent 2.4 GHz
Main memory	3,584 MB

Table Ⅱ: Specifications of video recording terminals

Component		Performance
OS	Client 1～4	Microsoft Windows 10 Pro Build 14393
CPU	Client1	Intel i7-6500U Equivalent 2.4 GHz
	Client2	Intel i7 2677M 1.8 GHz
	Client3	Intel i7 4650U 1.70 GHz
	Client4	Intel i3-4020Y 1.50 GHz
Main memory	Client1	8,118 MB
	Client2	3,999 MB
	Client3	8,097 MB
	Client4	3,999 MB

- The preprocessing time during which the client obtains the video data (equal to the time from the end of the previous frame data reception to the beginning of the next frame data transmission).

- The communication time while the different-world broadcasting server receives the frame data.

- The processing time on the different-world broadcasting server.

- The communication time during which the client receives the frame data from the different-world broadcasting server. The video processing time is defined as the time from the start of the video processing except for the image data reception time to the end of processing.

4.2 Evaluation Environment

In the evaluation, the different-world broadcasting server worked on the virtual machines provided by the Microsoft Azure service. Table 1 shows the specifications and OS of the virtual machines. We used five different virtual machines for the different-world broadcasting servers. Open CV, parallelized by Intel's parallel computing library TBB [10], was used as the library for executing video processing on the different-world broadcasting server. The video recording terminal is a PC installed in Osaka University. Table 2 shows the specifications of the video recording terminal PCs. 4 PCs were used, and the video recording terminal software were started two processes at a time, thereby operating the video recording terminal software of a total of four processes. These PCs accessed different world broadcasting servers via different home optical line networks to avoid networks (FLET'S Hikari by NTT, eo HIKARI by K-Opticom) congestion.

4.3 Evaluation Results

Figures 12 and 13 show the evaluation results for the turnaround time under the evaluation environment described in Section 4.2. The horizontal axis shows the recorded frame number, and the vertical axis the turnaround time in milliseconds. In Fig. 14 where the load is concentrated on a single different-world broadcasting server, the turnaround time increases in a nearly linear manner.

In Fig. 15, where the video effect processing request was distributed among four different-world broadcasting servers, the turnaround time increased in the latter half, since the calculation load increased. There were no sharp increases in the turnaround time compared with the result in Fig. 14. In addition, though the virtual machines of the Azure service shared the same performance parameters, the turnaround times fluctuated due to the communication delay between the video recording terminal and the different-world broadcasting servers, or other reasons. Over the first half of the frame numbers, the turnaround time in Fig. 15 gradually increased compared with the result in Fig. 14. This indicates that the concentration of video processing is suppressed. We also measured the turnaround time required to inquire the recommended different-world broadcasting server. The average value for a single inquiry, over 50 trials, was 16.28 msec.

The results revealed that the processing requests were allocated among different-world broadcasting servers based on the ECA rules, and the load could be distributed. In addition, we confirmed that, the turnaround times could fluctuate even if the hardware performance of the virtual machines was equal due to the influence of communication delay, etc.

5 DISSCUSSION

5.1 Evaluation Results

In the evaluation, the video processing sometimes concentrated on one server, even when the computational load was distributed to four servers. Moreover, the processing time increased when face was detected because we used actual video images and so it is difficult to grasp the overall pattern. Therefore, we draw the complimentary curves in the figures.

We used two networks for the evaluation. (FLET'S HIKARI by NTT, and eo HIKARI by K-Opticom). In case that the requests from video recording terminals were going to be concentrated on a single different-world broadcasting server, processing load was distributed to each different-world broadcasting server. Compared to the access time, the turnaround time was relatively long.

In the evaluation, video processing was detecting the face of a person in the video, with the designated effect inscribed in the ECA rule. As explained in the previous subsection, we gradually increased the load caused by human face detection and measured the turnaround time. The time needed to transmit and receive the frame data was compared by the turnaround time. The results are shown in Figs. 14 and 15.

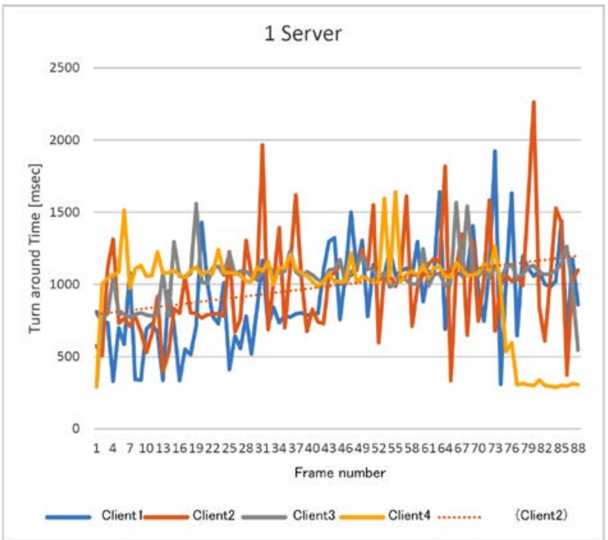


Figure 12: Turnaround times under one different-world broadcasting server

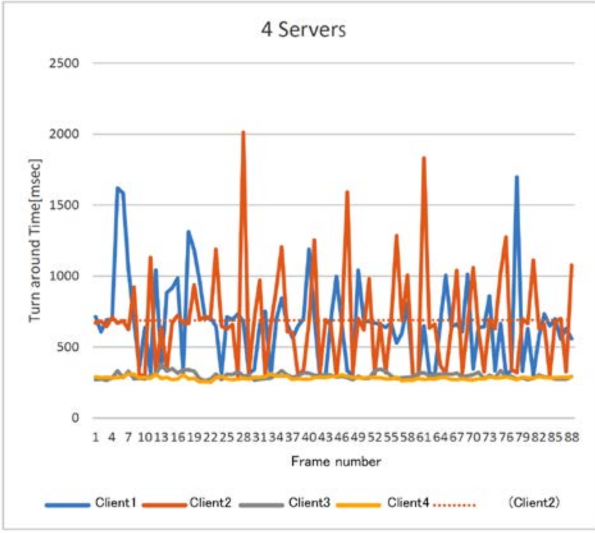


Figure 13: Turnaround times under four different-world broadcasting servers

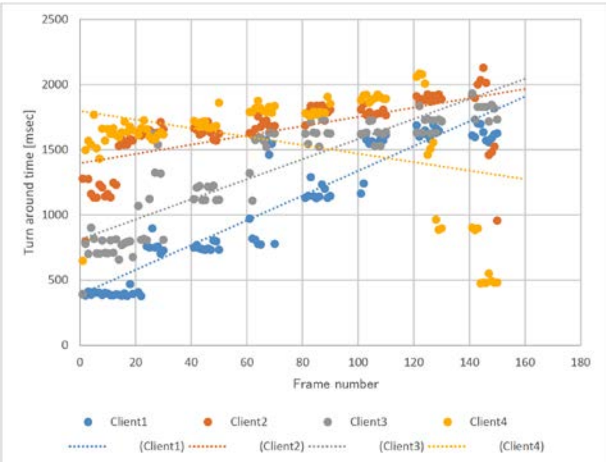


Figure 14: Turnaround times under one external different-world broadcasting server

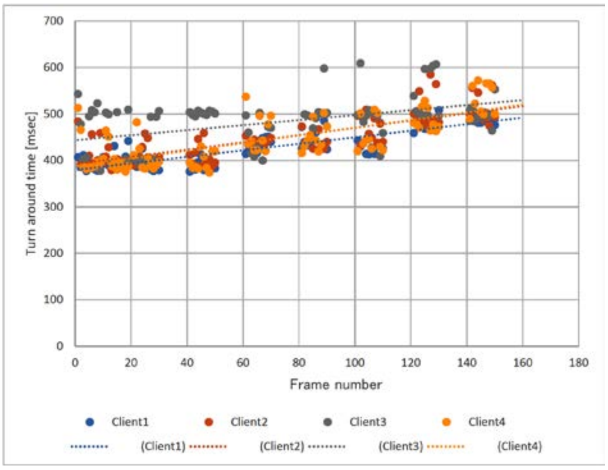


Figure 15: Turnaround times under four external different-world broadcasting servers

Fluctuations of the turnaround time were confirmed among the virtual machines of the cloud computing services. This arised by the factors such as fluctuations in the performance of the real servers in the cloud and differences in the network distances.

It is better for the users to consider these issues when they configure the system. Also, further improvement is required for ECA rule definitions.

5.2 ECA Rules Processing

Some video effects have sequences. For example, face detection processes are generally executed before the mosaic effects on the detected faces. `Timer' or `Message' functions of the ECA rules in our proposed system can define such sequences. When the sequences are defined in the ECA rules and there are order dependencies in the ECA rules, the system should execute the rules along with the order. Otherwise, when the ECA rules are order independent, the system can execute the rules in parallel and the processing time can be reduced compared with the ECA rules with order dependencies. Current system cannot process ECA rules in parallel and their parallel processing on cloud computing services is our future work.

6 CONCLUSION

In this research, we implemented and evaluated a distributed live Internet (different-world) broadcasting system using cloud services. By inscribing the processing assignment in ECA rules, it was possible to flexibly assign the virtual machine performance for video processing. In the implemented system, the PIAX platform was used for searching and communicating with the virtual machines, enabling continuous allocation of the video processing while live Internet broadcasting, even if the number of virtual machines changes. The system evaluation confirmed that the video processing turnaround time could be reduced by using our proposed system.

In the future, we plan to allocate processes with consideration of the hardware performance of different-world broadcasting servers, and distribute light weight processes on video recording terminals.

ACKNOWLEDGMENT

This research was supported by JSPS KAKENHI Grant Number JP15H02702 and JP17K00146, and part of this research is the result of NICT & Osaka University joint research "research and development of advanced network platform technology for large scale distributed computing".

REFERENCES

[1] S. Matsumoto, Y. Ishi, T. Yoshihisa, T. Kawakami, and Y. Teranishi, “Different Worlds Broadcasting: A Distributed Internet Live Broadcasting System with Video and Audio Effects,” in Proc. of 31st IEEE International Conference on Advanced Information Networking and Applications (AINA 2017), pp. 71-78 (2017).

[2] B. Cheng, “MediaPaaS: A cloud-based media processing platform for elastic live broadcasting,” Proceedings of the 7th IEEE International Conference on Cloud Computing (CLOUD 2014), pp. 713-720 (2014).

[3] Y. Gotoh, T. Yoshihisa, H. Taniguchi, and M. Kanazawa, “Brossom: a P2P streaming system for webcast,” Journal of Networking Technology, Vol. 2, No. 4, pp. 169-181 (2011).

[4] R. Roverso, R. Reale, S. El-Ansary, and S. Haridi, “Smooth-Cache 2.0: CDN-quality adaptive HTTP live streaming on peer-to-peer overlays,” Proceedings of the 6th ACM Multi-media Systems Conference (MMSys 2015), pp. 61-72 (2015).

[5] J. Dai, Z. Chang, and G.S.H. Chan, “Delay optimization for multi-source multi-channel overlay live streaming,” Pro-ceedings of the IEEE International Conference on Commu-nications (ICC 2015), pp. 6959-6964 (2015).

[6] T. Yoshihisa and S. Nishio, “A division-based broadcasting method considering channel bandwidths for NVoD services,” IEEE Transactions on Broadcasting, vol. 59, no. 1, pp. 62-71 (2013).

[7] D. Gibbon and L. Begaja, “Distributed processing for big data video analytics,” IEEE ComSoc MMTC E-Letter, vol. 9, no. 3, pp. 29-31 (2014).

[8] W.-C. Ting, K.-H. Lu, C.-W. Lo, S.-H. Chang, and P.C. Liu, “Smart video hosting and processing platform for Internet-of-Things,” Proceedings of the 2014 IEEE International Conference on Internet of Things (iThings 2014), pp. 169-176 (2014).

[9] M. Yoshida, T. Okuda, Y. Teranishi, K. Harumoto, S. Shimojyo, “PIAX: A P2P Platform for Integration of Multi-overlay and Distributed Agent Mechanisms,” Transactions of Information Processing Society of Japan / Information Processing Society of Japan, Vol. 49, No. 1, pp. 402-413, (2008).

[10] Thread Building Blocks, https://www.threadingbuildingblocks.org/, (referred October 1, 2017).

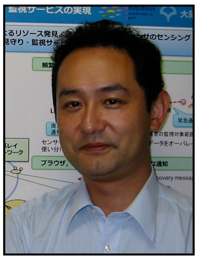
(Received October 20, 2017)

Satoru Matsumoto received his Diploma’s degrees from Kyoto School of Computer Science, Japan, in 1990. He received his Master’s degrees from Shinshu University, Japan, in 2004. From 1990 to 2004, he was a teacher in Kyoto School of Computer Science. From 2004 to 2007, he was Assistant Professor of The Kyoto College of Graduate Studies for informatics. From 2007 to 2010, he was Assistant Professor of Office of Society Academia Collabo-ration, Kyoto University. From 2010 to 2013, he was Assistant Professor of Research Institute for Economics & Business Administration, Kobe University. From 2015 to 2016, he was a specially appointed assistant professor of Cybermedia Center, Osaka University. From April 2016 to September 2016, he became a specially appointed researcher. Since November 2016, he became an assistant professor. His research interests include distributed processing systems, rule-based systems, and stream data processing. He is a member of IPSJ, IEICE, and IEEE

Yoshimasa Ishi received his B.E. degrees from Kyoto Institute of Technology, Japan, in 2004 and his M.I. degrees from Osaka University, Japan, in 2006, respectively. From 2006 to 2008, and from 2012 to 2015, he was a specially appointed researcher of Cybermedia Center, Osaka University. From 2008 to 2012, he was a specially appointed researcher of Graduate School of Information Science and Technology, Osaka University. Since January 2017, he has been a specially appointed researcher of Institute for Datability Science, Osaka University. His research interests include technologies for distributed network systems and its development. He is a member of the IPSJ.J.

Tomoki Yoshihisa received the Bachelor's, Master's, and Doctor's degrees from Osaka University, Osaka, Japan, in 2002, 2003, 2005, respectively. Since 2005 to 2007, he was a research associate at Kyoto University. In January 2008, he joined the Cybermedia Center, Osaka University as an assistant professor and in March 2009, he became an associate professor. From April 2008 to August 2008, he was a visiting researcher at University of California, Irvine. His research interests include video-ondemand, broadcasting systems, and webcasts. He is a member of the IPSJ, IEICE, and IEEE.

Tomoya Kawakami received his B.E. degree from Kinki University in 2005 and his M.I. and Ph.D. degrees from Osaka University in 2007 and 2013, respectively. From 2007 to March 2013 and from July 2014 to March 2015, he was a specially appointed researcher at Osaka University. From April 2013 to June 2014, he was a Ph.D. researcher at Kobe University. Since April 2015, he has been a assistant professor at Nara Institute of Science and Technology. His research interests include distributed processing systems, rule-based systems, and stream data processing. He is a member of the IPSJ and IEEE.



Yuuichi Teranishi received his M.E. and Ph.D. degrees from Osaka University, Japan, in 1995 and 2004, respectively. From 1995 to 2004, he was engaged Nippon Telegraph and Tele-phone Corporation (NTT). From 2005 to 2007, he was a Lecturer of Cybermedia Center, Osaka University. From 2007 to 2011, He was an associate professor of Graduate School of Information Science and Technology, Osaka University. Since August 2011, He has been a research man-ager and project manager of National Institute of Information and Communications Technology (NICT). He received IPSJ Best Paper Award in 2011. His research interests include technologies for distributed network systems and applications. He is a member of IPSJ, IEICE, and IEEE.

Regular Paper

Real-time Performance of Embedded Platform for Autonomous Driving Using Linux and a GPU

Shota Mizuno†, Shinya Honda‡*, Ryo Kurachi‡**, and Naoya Chujo***

†Graduate School, Aichi Institute of Technology, Japan
‡Graduate School, Nagoya University, Japan
***Aichi Institute of Technology, Japan
{†b16727bb,***ny-chujo}@aitech.ac.jp
*honda@ertl.jp
**kurachi@nces.is.nagoya-u.ac.jp

Abstract - Autonomous driving systems and advanced driver assistance systems (ADAS) have been developed to improve driving safety. These systems are becoming increasingly complicated, and the computing power they require continues to increase. Software platforms, such as Automotive Grade Linux, are being developed to handle the complexity of autonomous driving, and graphical processing units (GPUs) are being used in these systems for tasks such as recognition of driving environments, deep learning, and localization. However, to the best of our knowledge, the real-time performance of Linux combined with a GPU for ADAS has not yet been reported.

In this study, we developed lane keeping and collision avoidance systems to evaluate the real-time performance of Linux and a GPU. The experimental results show that the real-time performance of the system could be improved using widely available versions of Linux without software customization. Also, although the GPU execution speed was sufficiently high, camera image capture was relatively slow and created a bottleneck in the system.

Keywords: Autonomous driving, Advanced driver assistance systems, Linux, Real-time performance, GPU

1 INTRODUCTION

In recent years, autonomous driving systems and advanced driver assistance systems (ADAS) have been developed to improve driving safety [1], [2]. In fact, the first cars equipped with autonomous driving level 3, as defined by the Society of Automotive Engineers [3], is scheduled to be released in 2018 [4]; this level allows the vehicle to assume control of safety-critical functions. In both autonomous driving systems and ADAS, a large number of sensors, such as cameras, radar, light detection sensors, and ultrasonic sensors, are used to recognize various aspects of the driving environment [5]. The driving system must be capable of simultaneously processing a large amount of information from these sensors in real time in order to operate at highway speeds and respond to the environment on any roadway. For this reason, two challenges are common to these systems: complexity of the software and demand for computing power.

Regarding software complexity, standard software platforms are already being developed. For instance, Automotive Grade Linux [6] is being developed by many automakers and suppliers for in-vehicle infotainment systems. The abundant device drivers, application programming interfaces, and libraries within Linux are expected to reduce the development cycle and cost of new systems. Furthermore, there have been efforts to improve the real-time performance of Linux using RTLinux [7], which is a Linux kernel augmented by a dedicated patch. Here, real-time is defined as a time constraint for processing by a specified deadline, and it is the most important property required for in-vehicle systems.

To increase the computational speed of ADAS hardware, GPUs are increasingly being adopted. GPUs are necessary for various autonomous driving system tasks, such as recognition of vehicles and pedestrians using advanced image processing, deep learning, and localization. The development of semiconductor technology has dramatically accelerated the advancement of GPUs and given them a massively parallel architecture. In particular, GPUs seem to be essential for improving the recognition performance of ADAS using deep learning.

However, to the best of our knowledge, evaluation of the real-time performance of Linux and GPUs for autonomous driving systems has not yet been reported, despite its importance. In our study, two autonomous driving platforms were considered [8]. One platform was for evaluation of the real-time performance of an autonomous system using Linux, and the other platform was used to evaluate the performance of Linux with a GPU. We examined the factors affecting the real-time performance of systems developed on these platforms. In addition, the real-time processing capabilities of the embedded GPU were verified for the second platform. By evaluating these platforms, we aim to obtain guidelines for implementing software for autonomous driving systems and ADAS. Another objective is to evaluate the potential of embedded GPUs and understand processing bottlenecks.

This paper is organized as follows. Section 2 describes the two platforms based on the general architecture for autonomous driving systems, describes the elements comprising these platforms, and explains the systems used for evaluation. Section 3 discusses experiments conducted to evaluate the platform using Linux and a real-time operating system (RTOS) and determine its real-time performance. In Section 4, we present experiments using a platform based on Linux,

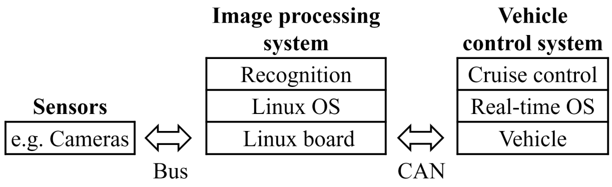


Figure 1: Minimum architecture for autonomous driving system

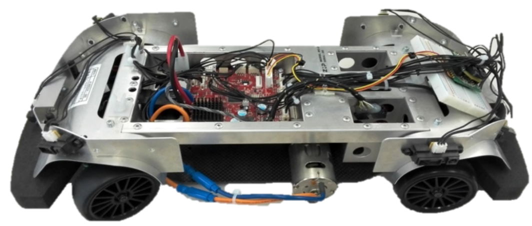


Figure 2: RoboCar®1/10 for AP

the RTOS, and a GPU. Experimental considerations are discussed in Section 5, and Section 6 provides conclusions.

2 PLATFORMS FOR AUTONOMOUS DRIVING

In this section, the evaluation platforms for autonomous driving are explained.

2.1 Minimum Architecture for Autonomous Driving

Autonomous driving systems and ADAS normally contain a number of different sensors, most commonly onboard cameras, which are indispensable for recognizing lanes, pedestrians, and signs. Use of open-source libraries enables reduction of cost and development cycle, and for image processing, function libraries such as OpenCV [9] are available for Linux. Therefore, we decided to use Linux for image processing in both platforms.

Figure 1 shows the minimum architecture for an autonomous driving system, which consists of three parts, an onboard camera, an image processing system, and a vehicle control system. On the basis of this architecture, two platforms were developed for this study. The features of these two platforms are given in Table 1.

Both platforms A and B use Linux and an RTOS, but only Platform B is equipped with a GPU. In Platform A, relatively lightweight processing is performed, whereas in Platform B, advanced image processing requiring substantial computing power is performed.

2.2 RoboCar®1/10 for AP

RoboCar®1/10 for automotive platform (AP) (hereafter "RoboCar") [10], shown in Fig. 2, is a 1:10 scale miniature version of an actual car. It has a Renesas V850 central processing unit (CPU) with a TOPPERS/ATK2 OS [11] RTOS, which was developed for automotive systems. In this study, RoboCar provided the base platform, and its CPU was used for the vehicle control system. The basic configuration of RoboCar is given in Tab. 2.

2.3 Platform A Using Linux

This section introduces the platform using Linux. The architecture developed for Platform A is shown in Fig. 3.

A Raspberry Pi 2 Model B was adopted as the image processing system. This system consists of a small single-board computer that runs a Linux-based OS and has a quad-core Arm Cortex-A7 processor and 1 GB of memory (DDR-SDR-

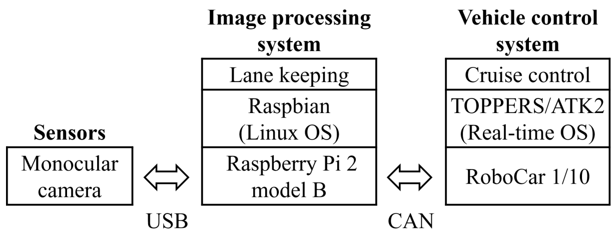


Figure 3: Architecture of Platform A

AM). The Raspberry Pi and Robocar were connected via a controller area network (CAN). A monocular camera (iBUFFALO BSW- 20KM11) with a resolution of 640 × 480 and a maximum frame rate of 30 frames per second (fps) was connected to the image processing system with a universal serial bus (USB). Figure 4 is a photograph of Platform A.

2.4 Platform B Using Linux with GPU

This section introduces the platform equipped with a GPU running Linux. The architecture of this platform is shown in Fig. 5.

In this platform, an NVIDIA Jetson TX1 module [12] was used for image processing to execute more demanding stereographic image processing. The Jetson TX1 is a card-sized board with an NVIDIA Tegra X1 chip running at 10 W. The chip has 256 CUDA cores for the GPU and a quad-core Arm Cortex-A57 CPU with a 2-MB L2 cache and a 4-GB random access memory (LPDDR4). The Jetson TX1 OS is Linux for Tegra (L4T) R23.2 provided by NVIDIA. Figure 6 shows the Jetson TX1 module, and Fig. 7 is a photograph of Platform B.

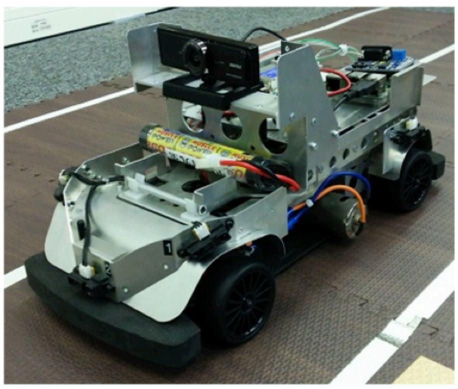


Figure 4: Platform A: RoboCar with monocular camera and Raspberry Pi

Table 1: Features of the two platforms

System	Platform A	Platform B
Sensors	iBUFFALO BSW20KM11 monocular camera	Stereolabs ZED stereo camera
Image processing	Raspberry Pi 2 Model B CPU: 4× Arm Cortex-A7 GPU: none	NVIDIA Jetson TX1 CPU: 4× Arm Cortex-A57 GPU: 256× CUDA core
Vehicle control	ZMP RoboCar®1/10 for AP	

Table 2: Basic configuration of RoboCar

Dimensions	429×195×212 mm
Weight	1.8 kg
Internal sensors	Gyro Accelerometer 5× Rotary encoder
External sensors	8× Infrared range-finding sensor
Battery	7.2-V NiMH battery
CPU	Renesas V850
OS	TOPPERS/ATK2

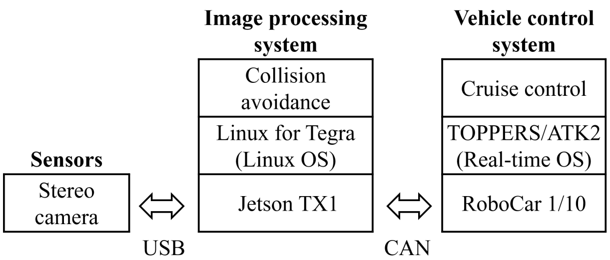


Figure 5: Architecture of Platform B

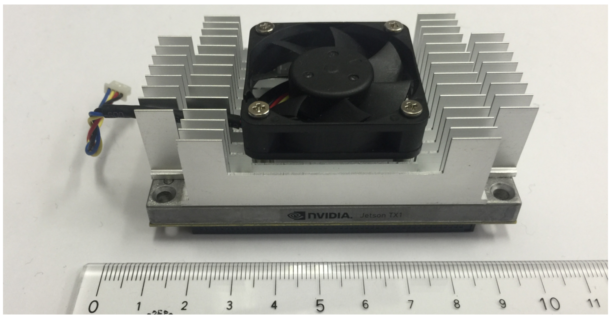


Figure 6: NVIDIA Jetson TX1 (with centimeter scale)

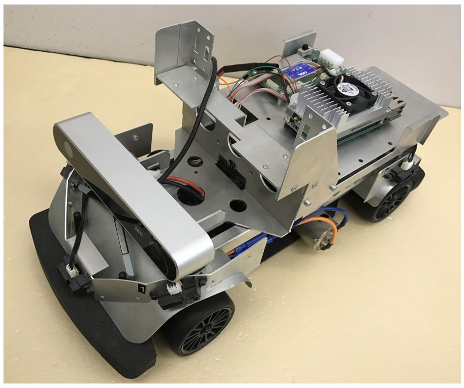


Figure 7: Platform B: RoboCar with stereo camera and Jetson TX1 GPU

Table 3: Key features of ZED Stereo Camera

Dimensions		175×30×33 mm
Lens	**Field of view**	110°
	F-number	2.0
Depth	**Depth range**	0.5 20 m
	Stereo baseline	120 mm
Resolution	**HD1080**	1920×1080 / 30Hz
	HD720	1280×720 / 60 Hz
	WVGA	672×376 / 100Hz

In this setup, the Jetson TX1 module and RoboCar were connected via CAN using an Auvidea J120 carrier board [13]. A USB 3.0 Stereolabs ZED stereo camera was connected to the Jetson TX1 board; this camera obtains images useful for both image processing tasks evaluated in this study. Key features of ZED stereo camera are shown in Fig. 3.

Three of this camera's image resolutions were used in this study: wide video graphics array (WVGA) (672 × 376), HD720 (1,280 × 720), and HD1080 (1,920 × 1,080).

2.5 CUDA

The Compute Unified Device Architecture (CUDA) is a parallel programming language and model developed by NVIDIA [14], [15] that is available in the Jetson TX1 module. CUDA is supported by OpenCV, a library for computer vision functions that was used to develop the image processing program for both platforms.

In CUDA, the TX1 CPU is treated as the host and its GPU as the device. The term "kernel" refers to functions that run on the device, and a kernel consists of numerous "threads." All threads run the same code, and in image processing, one

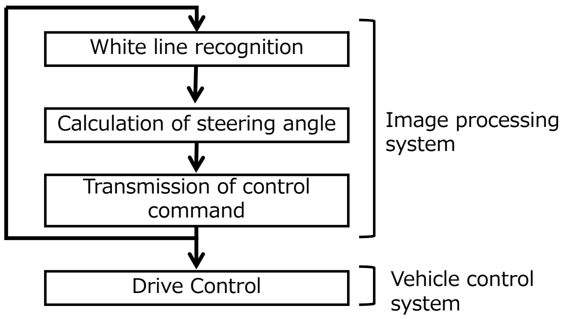


Figure 8: Flow diagram of lane keeping system

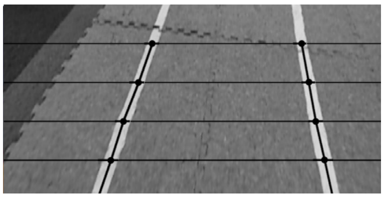


Figure 9: Example of white line recognition

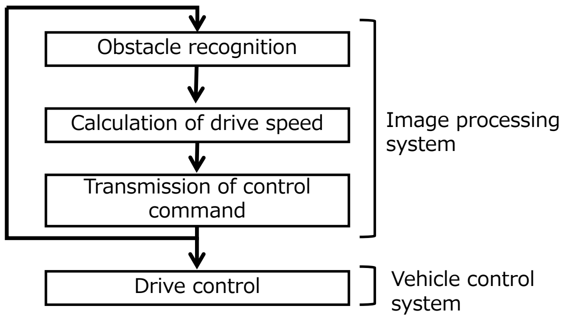
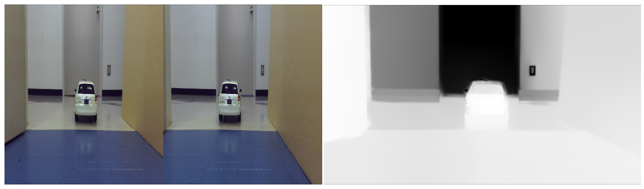


Figure 10: Flow diagram of collision avoidance system



(**a**) Stereo image (**b**) Depth map

Figure 11: Example of obstacle recognition with a stereo camera

thread corresponds to one pixel. First, a program executed on the host side is activated. Next, the device loads the kernel program, and the host passes the generated data to the device and "kicks" it (to instruct the host to start the kernel). The device executes the kernel and asynchronously returns the obtained result to the host. The processing on the GPU side can be synchronized, but in this study, unless otherwise noted, processing on the GPU side was executed asynchronously.

2.6 Lane Keeping System

Lane keeping [16] is a basic driving support system, and its configuration is simple. A lane keeping system consisting of white line recognition and steering control was implemented in Platform A to verify the real-time performance of Linux. Figure 8 is a flow diagram of the lane keeping system, and Fig. 9 shows an example of white line recognition.

The camera image is used by the image processing system to recognize white lines. Then, the image processing system uses the recognition result to calculate the steering angle required to run along the white line and transmits it to the vehicle control system.

In this study, the target time for white line recognition was set to 30 ms, which is generally regarded as real-time performance.

2.7 Collision Avoidance System

Collision avoidance systems are being adopted by many manufacturers. The sensors used in collision avoidance systems include millimeter-wave and infrared radar sensors and monocular and stereo cameras. We adopted collision avoidance as the task for evaluating Platform B using the GPU because collision avoidance requires more computing power, in comparison to that needed for lane keeping. Figure 10 is a flow diagram of the collision avoidance system.

In this system, the GPU generates a depth map from the stereo image and calculates the driving speed, which is transmitted to the vehicle control system via the CAN. When an obstacle is recognized in the depth map, the vehicle is stopped to avoid collision. An example of obstacle recognition is shown in Fig. 11, where the depth map expresses nearer as brighter.

It is said that "assuming car speed of 80 Km/Hr, the stopping distance reduces from 55m to 45m as frame-rate goes from 10fps to 15fps (for systems as of today) to 30 fps (for system in future)." [17]. In this study, the target time of collision avoidance system was set to 33 ms which is calculated from 30 fps.

3 PLATFORM A: REAL-TIME PERFORMANCE WITH LINUX

In this section, we report the evaluation of the real-time performance of Linux used for lane keeping with Platform A.

3.1 Lane Keeping System on Platform A

The iBUFFALO BSW20KM11 monocular camera used for sensing in Platform A was attached to the top of the RoboCar and output VGA (640 $\times$ 480) images at 30 fps. One cycle of image processing and response is shown in Fig. 12.

In the experiments, Platform A navigated a round course created using a tile mat and white tape. The lane keeping processing time was measured for each cycle, which started with image capture and ended with transmission of steering

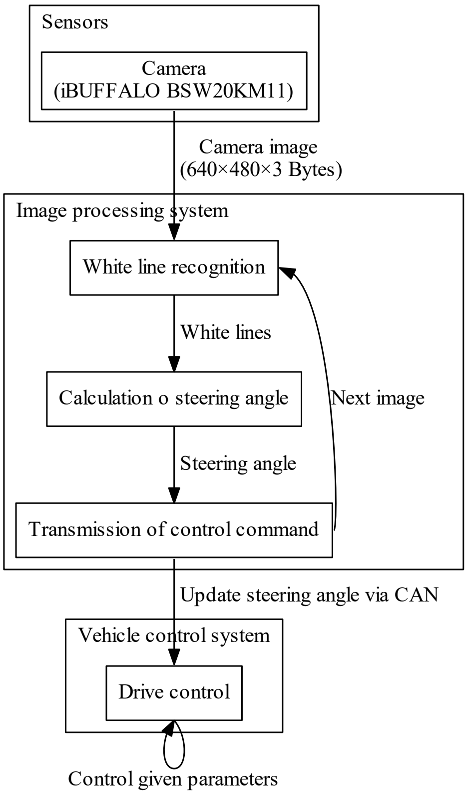


Figure 12: Process flow in lane keeping system on Platform A

instructions via CAN. The clock rate of the Raspberry Pi was set to 900 MHz.

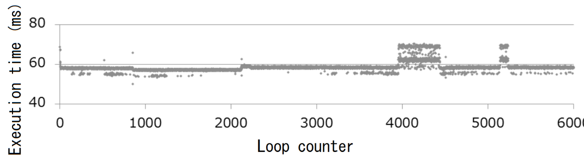
3.2 Processor Affinity for CPU Selection

Processor affinity is a property used to specify which CPU executes each process. In Linux, the OS normally automatically assigns an execution processor based on this property and the priority of the process. If the process can be executed by another processor, the execution may sometimes be reassigned, which delays the process while the cache is copied.

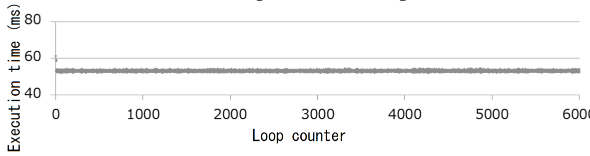
In this study, the affinity was set to execute the lane keeping program on core 2 of the Raspberry Pi using the taskset command in Linux. The execution times with and without the specification of the execution processor were measured for 10,000 frames. The results are shown in Fig. 13.

As shown in Fig. 13a, when the execution processor was not specified, lane keeping was typically executed within 60 ms, although it sometimes increased to approximately 70 ms. We investigated the cause and found that the occasional delay was caused by neither dynamic clock changes, a temperature rise, nor the software algorithm. With these factors ruled out, we concluded that the delay was caused by the reassignment of the execution processor. As shown in Fig. 13b, this delay is suppressed when the execution processor is specified.

The average execution time and worst-case execution time (WCET) are given in Tab. 4. The average execution time



(a) Execution processor not specified



(b) Execution processor specified

Figure 13: Execution time with and without specification of the execution processor

Table 4: Average execution time and WCET with and without specification of execution processor

Execution time	Not specified	Specified
Average	59.4 ms	53.1 ms
WCET	78.1 ms	61.3 ms

and the WCET were both lower when the execution processor was specified than when it was not specified, demonstrating that the WCET can be improved by specifying the execution processor.

During normal processing in a Linux OS, a timer interrupt occurs at regular intervals. When the interrupt occurs, the scheduler examines the process state and determines the priority of the processes to be executed. Because the priority of a process that is using a CPU for a long time decreases with time, the timer interrupt enables the execution of other waiting processes. With this mechanism, Linux maintains the even execution of processes.

When lane keeping was executed without options, the priority for this process is the same as that for other processes. Therefore, lane keeping execution may occasionally be postponed by the scheduler. In addition, if the execution time of one cycle of the process is long, the scheduler is activated during the execution, and other higher-priority processes are executed. Then, the original process is delayed until it becomes executable again. This is thought to be the reason for the occasional execution delay when no processor is specified in Linux.

3.3 Real-time Process and RTLinux

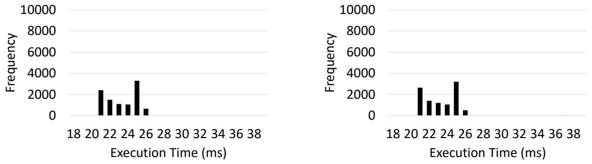
Next, we enhanced Linux for real-time performance and evaluated the results. RTLinux was developed to improve the real-time performance of Linux. In the real-time process, the execution of the processor is given higher priority.

RTLinux is obtained by augmenting the Linux kernel by applying the RT-Preempt patch [18] provided by the Linux community. Various other patches exist for this purpose, including Xenomai and real-time application interface [19], but these were not used in this study.

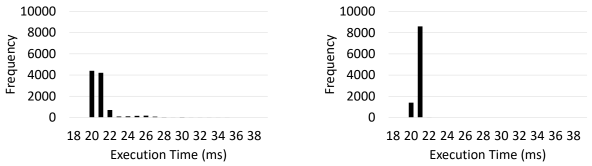
RTLinux was implemented on Platform A, the lane keeping experiment was repeated, and the normal and real-time exe-

Table 5: Kernel and process conditions for each real-time (RT) case

Application	a	b	c	d
Kernel	Normal	Normal	RT	RT
Process	Normal	RT	Normal	RT



(**a**) Linux, normal process (**b**) Linux, real-time process



(**c**) RTLinux, normal process (**d**) RTLinux, real-time process

Figure 14: Histograms of lane keeping execution times for four real-time cases

cution times were compared. When using RTLinux, the time required for the camera to capture the image was abnormally long. This was likely because the device driver was updated when the RT- Preempt patch was applied. Therefore, the time required for image capture was not included in the execution time. Experiments were conducted for four cases with each possible combination of real-time or normal kernels and real-time or normal processes. The conditions for the four cases are given in Tab. 5.

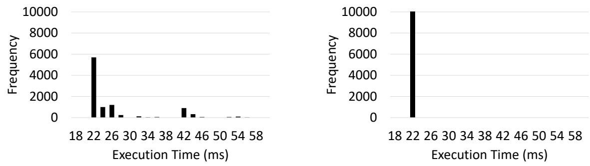
Histograms of the lane keeping execution times obtained for all four cases are shown in Fig. 14, and the corresponding average execution times and WCETs are given in Tab. 6.

When the program was executed as a normal process on RTL- inux, the execution time was unstable in comparison with the normal process executed on the normal Linux kernel. However, when it was executed as a real-time process on RTLinux, the execution time was stable, and the delay was greatly suppressed because the delay for dispatching the process is decreased by RTLinux.

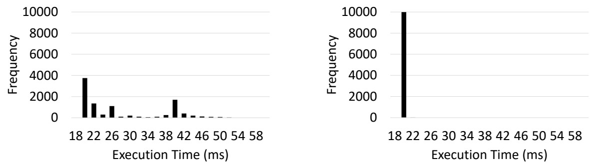
Applying the RT-Preempt patch reduced the time to start the scheduler. The real-time performance of the lane keeping program deteriorated in the kernel with the RT-Preempt patch applied without options. When the process was executed as a normal process, the execution state is thought to switch more frequently. When the process was executed as a real-time process on RTLinux, it was preferentially executed by one of the cores, even if the execution core was not specified. The real-time performance was also improved in normal Linux by executing it as a real-time process.

Table 6: Average execution times and WCETs for four lane keeping real-time cases

Execution time	a	b	c	d
Average	26.8 ms	20.5 ms	20.5 ms	20.3 ms
WCET	40.5 ms	21.5 ms	52.2 ms	21. 7ms



(**a**) Linux, normal process (**b**) Linux, real-time process



(**c**) RTLinux, normal process (**d**) RTLinux, real-time process

Figure 15: Histograms of lane keeping execution time for four real-time cases with CPU load

Table 7: Average execution times and WCETs for four lane keeping real-time cases with CPU load

Execution time	a	b	c	d
Average	25.9 ms	20.4 ms	27.4 ms	19.1 ms
WCET	67.4 ms	21.4 ms	57.3 ms	20.6 ms

3.4 Real-time Process and RT Linux with CPU Load

In the experiments discussed so far, only a single user process was executed. However, in an actual system, multiple processes are executed simultaneously. Therefore, the delay in executing processes was examined while applying a background processing load. For the same four cases listed in Tab. 5, a CPU load was applied using the stress command and processing was performed for 10,000 periods. Histograms of the lane keeping execution times are shown in Fig. 15 for the four real-time cases with CPU load, and the corresponding average execution times and WCETs are given in Tab. 7.

According to Figs. 15a and 15c, the fluctuation of the execution cycle was larger in both kernels when the process was executed as a normal process. As shown in Tab. 7, the average execution time for the lane keeping program executed as a normal process in the normal Linux kernel (case a) was 25.9 ms, and the WCET was 67.4 ms. In this case, the variation in the execution cycle was not evenly distributed but was concentrated in two ranges: 22–26 ms and 42–44 ms. In contrast, when executed as a real-time process in the normal Linux kernel (case b), the average execution time and WCET were 20.4 ms and 21.4 ms, respectively. Thus, executing the program as a real-time process stabilized the execution time.

When a CPU load was applied and the program was executed as a normal process, the execution cycle greatly fluctuated in both kernels. However, when it was run as a real-time process with an applied CPU load, the execution time was stabilized for both kernels. The execution time was stabilized by assigning the process a high priority; however, since an actual system contains multiple high-priority processes, it is a matter which process is prioritized.

4 PLATFORM B: REAL-TIME PERFORMANCE USING GPU WITH LINUX

This section discusses the evaluation of the real-time performance with Platform B using Linux and a GPU.

4.1 Lane Keeping System on Platform B

Image processing for the lane keeping program was implemented on the Jetson TX1 module in place of the Raspberry Pi; however, lane keeping was found to be too lightweight to use the GPU processor. The execution time for the lane keeping program was 5.4 ms (excluding image capture time) when using the TX1 CPU, and the time increased to 8.1 ms when using the TX1 GPU. Data transfer between the CPU and GPU occupied 1.7 ms.

Because lane keeping is a lightweight process and the overhead of the data transfer is larger than the execution time reduction achieved by using the GPU, the TX1 CPU executed the lane keeping task on Platform B.

4.2 Collision Avoidance System on Platform B

A stereo camera was adopted for the collision avoidance system because it is the sensor that is best able to both detect the lane and identify obstacles. The collision avoidance system configuration using the stereo camera and Jetson TX1 module is shown in Fig. 16.

When capturing WVGA image, the image data is captured by processor of image processing system. The depth map (672×376 ×1 Bytes) is generated and used on GPU to calculate the drive speed of RoboCar. Finally, drive speed is send to vehicle control system via CAN.

In this experiment, the RoboCar was located on a miniature straight course. By bringing an obstacle closer, we confirmed that the collision avoidance system was working. The image processing time for collision avoidance was measured for each cycle, which began with image capture and ended with transmission of instructions via the CAN. The Jetson TX1 clock rate of was 1.9 GHz. The experiment was performed with three different resolutions.

4.3 Dependence on Image Resolution

Simple pedestrian detection has already been put to practical use, but high-resolution images will be necessary to improve the detection distance and estimation of pedestrian movement direction. When 8-mega-pixel images are used, it is possible to detect pedestrians at distances of up to 200 m [17].

Figure 17 shows the processing speeds for obstacle recognition experiments conducted using stereo images at three

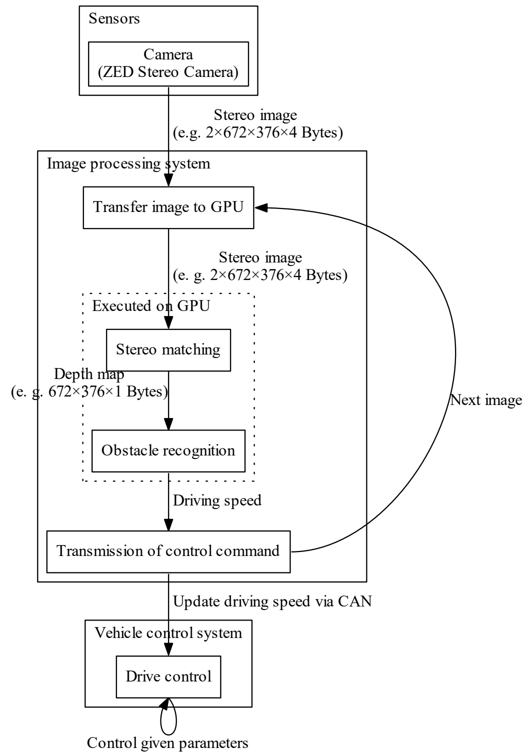


Figure 16: Process flow in collision avoidance system on Platform B

Table 8: Resolution-dependent stereo image sizes and processing speeds for WVGA

Resolution	Image size	Processing speed
WVGA	1.0×	1.0×
HD720	3.7×	2.2×
HD1080	8.3×	4.0×

resolutions. The processing speeds for WVGA, HD720, and HD1080 were 68, 31, and 17 fps, respectively. Table 8 gives the processing speeds for the three resolutions relative to that of WVGA.

Although the image sizes at resolutions of HD720 and HD-1080 were, respectively, 3.7 and 8.3 times that of WVGA, the processing speeds were only 2.2 and 4.0 times that of WVGA, respectively. This shows that the processing parallelization in the GPU was utilized. However, most of this execution time was spent on capturing the camera image. Figure 18 shows the breakdown of the synchronously measured execution time of the collision avoidance system.

In the case of WVGA, the CPU execution time for one frame was approximately 15 ms, and approximately 10 ms of that time was spent on image capture. The GPU completed processing in 8 ms, and this processing time did not increase much at higher resolutions. Thus, processing in the TX1 CPU, particularly with regard to the capture of camera images, is a performance bottleneck.

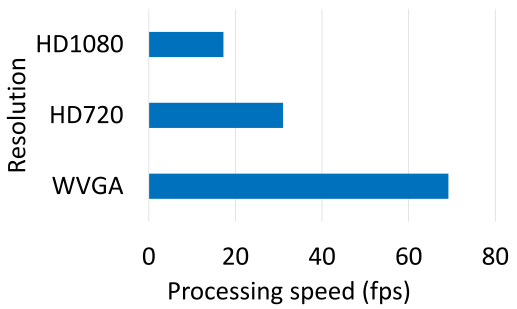


Figure 17: Comparison of stereo image processing speeds at three different resolutions

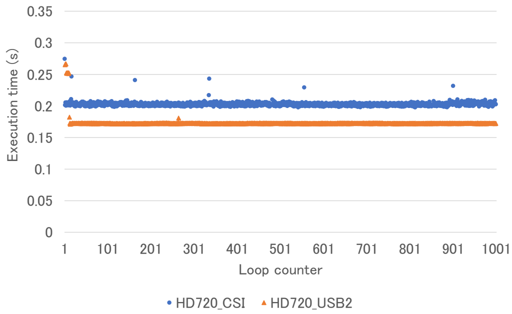


Figure 20: Comparison of capture execution time for USB and CSI camera interfaces

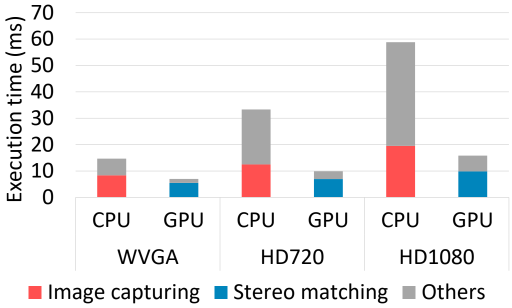
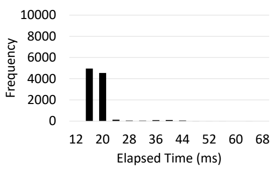


Figure 18: Execution time breakdown for collision avoidance system

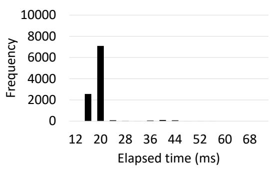
4.4 Processor Affinity for CPU Selection

Experiments were also conducted to determine the processor affinity for Platform B. The recognition process was fixed to core 2 of the Jetson TX1 board, and the program was run as both with and without processor specification. The elapsed time per processing cycle was measured since processing on GPU performed asynchronously. The resulting elapsed time histograms are shown in Fig. 19.

The elapsed times were concentrated near 16 and 20 ms for both the normal and real-time processes. In addition, when the collision avoidance program was executed as a real-time process, the elapsed time increased to nearly 60 ms in approximately 0.02% of the frames. This is because the program used in the experiment consisted of several subprocesses and sometimes other subprocesses were prioritized, causing execution to be postponed.



(a) Execution processor not specified



(b) Execution processor specified

Figure 19: Histogram of the collision avoidance elapsed time with and without specification of the execution processor

4.5 Camera Interfaces

In the experiments so far, cameras were connected via USB, and these cameras were the bottleneck in the system. Therefore in this section, we compared the USB camera interface and camera serial interface (CSI) [20].CSI is a camera/processor interface that is commonly used in embedded systems.

We measured and compared the time for capturing ZED stereo camera images between the USB and CSI camera interfaces with the Raspberry Pi 2. Figure 20 shows the execution time for capturing HD720 images for each interface.

Average execution times using the CSI and USB interfaces were 0.2 s and 0.17 s, respectively. Thus, in this experiment, use of the CSI interface did not speed up image capture.

5 CONSIDERATIONS

The experimental results obtained in this study show that the real-time performance of autonomous driving software programs implemented in Linux can be improved by specifying the execution core without optimizing the programs. Executing programs as real-time processes was also effective in both the normal Linux and RTLinux kernels. Furthermore, by applying the RT patch to the Linux kernel, it was possible to stabilize the execution time.

By executing the top-priority process as a real-time process, real-time performance can be secured even in Linux, which is a general-purpose OS. Therefore, it is expected that the wealth of software assets developed in Linux can be used, which leads to a reduction of development cycle time and cost.

It was verified that these methods are effective when executing one process, but an actual system will likely host multiple high-priority processes. Therefore, it is necessary to design the priorities of processes and system calls and verify the system performance with sufficient test vectors.

With regard to the GPU hardware used for image processing system in these experiments, the operation executed on the GPU side was sufficiently fast and the real-time performance was considered to be satisfactory. As GPU technology continues to develop, the image resolution and frame rate obtained using a GPU are expected to improve. However, the transfer time between the CPU and GPU on the TX-1 board and the image capture process remain as bottlenecks. Without an improved high-speed transfer channel and a high-speed interface with the camera, it is difficult to take advantage of the GPU's potential.

For comparatively lightweight computations, such as white line recognition, processing can be performed using only a CPU as in Platform A. The GPU usage method adopted for Platform B makes it possible to perform basic arithmetic operations with only the CPU.

6 CONCLUSION

In this study, we developed two platforms for autonomous driving and evaluated the real-time performance of image processing for lane keeping and collision avoidance systems using Linux. We found that allowing Linux to assign the processor decreased the real-time performance of both systems. Using the Linux taskset command to assign execution to a specific processor avoided the delay caused by the cache copying required for reassignment. We also found that executing the process as a real-time process on RTLinux can stabilize the execution time even when other processes are running. Additionally, we found that an embedded GPU provided process execution at sufficiently high speeds, but the overall speed was constrained by the speed of camera image capture, which was a major performance bottleneck in the system.

Future work will include considering a more complex system, such as a robot operating system, to verify the performance of a practical autonomous driving system using the advanced features evaluated here.

ACKNOWLEGMENTS

The authors thank Zenzo Furuta (Graduate school, Aichi Institute of Technology, until May 2017) for experimental support.

REFERENCES

[1] C. Urmson, et al. "Autonomous driving in urban environments: Boss and the urban challenge," Journal of Field Robotics Vol. 25, No. 8, pp. 425-466, (2008).

[2] A. Lindgren and F. Chen, "State of the art analysis: An overview of advanced driver assistance systems (adas) and possible human factors issues," Human factors and economics aspects on safety pp. 38-50, (2006).

[3] SAE International, "AUTOMATED DRIVING", http://www.sae.org/misc/pdfs/automated_driving.pdf (referred October 10, 2017).

[4] Audi newsroom, "Audi piloted driving", https://media.audiusa.com/models/piloted-driving (referred October 10, 2017).

[5] J. Levinson, et al. "Towards fully autonomous driving: Systems and algorithms," Intelligent Vehicles Symposium (IV), 2011 IEEE, pp. 163-168, (2011).

[6] Linux Foundation, "AUTOMOTIVE GRADE LINUX", https://www.automotivelinux.org/ (referred May 27, 2017).

[7] Linux Foundation, "Real-Time Linux", https://wiki.linuxfoundation.org/realtime/start.html (referred May 27, 2017).

[8] S. Mizuno, S. Honda, R. Kurachi, D. Hayakawa and N. Chujo, "Evaluation of Embedded Platform for Autonomous Drive using Linux and GPU," Proceedings of International Workshop on Informatics, pp. 53-58, (2017).

[9] G. Bradski and Adrian Kaehler, "Learning OpenCV: Computer vision with the OpenCV library," O'Reilly Media, Inc., (2008).

[10] ZMP Inc, "1/10 scale RoboCar(r) 1/10", http://www.zmp.co.jp/products/robocar-110 (referred May 27, 2017).

[11] TOPPERS Project, "TOPPES/ATK2", https://www.toppers.jp/atk2.html (referred May 27, 2017).

[12] NVIDIA, "Embedded Systems", http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html (referred October 10, 2017).

[13] AUVIDEA, "j120", https://auvidea.com/j120/ (referred October 10, 2017).

[14] NVIDIA, "CUDA Zone", https://developer.nvidia.com/cuda-zone (referred May 27, 2017).

[15] D. Kirk, "NVIDIA CUDA software and GPU parallel computing architecture," ISMM, Vol.7, pp.103-104, (2007).

[16] R. Bosch, "BOSCH Automotive Handbook," the 8th Edtion, (2011).

[17] EETimes, "ADAS Front Camera: Demystifying Resolution and Frame-Rate", http://www.eetimes.com/author.asp?section_id=36&doc_id=1329109 (referred May 27, 2017).

[18] S. Arthur, C. Emde, and N. Mc Guire, "Assessment of the realtime preemption patches (RT-Preempt) and their impact on the general purpose performance of the system," Proceedings of the 9th Real-Time Linux Workshop, (2007).

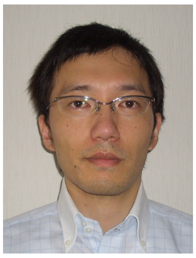
[19] A. Barbalace, et al. "Performance comparison of VxWorks, Linux, RTAI, and Xenomai in a hard real-time application", IEEE Transactions on Nuclear Science Vol. 55, No. 1, pp. 435-439, (2008).

[20] MIPI Alliance, "Camera and Imaging", https://mipi.org/specifications/camera-and-imaging (referred October 10, 2017).

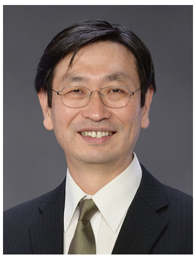
(Received October 20, 2017)
(Revised December 28, 2017)

Shota Mizuno received the B.I. degree in Information Science from Aichi Institute of Technology, Japan in 2016. He is now a master course student of Aichi Institute of Technology, Japan. His research interest includes embedded system and automotive electronics.

Shinya Honda received his Ph.D. degree in the Department of Electronic and Information Engineering, Toyohashi University of Technology in 2005.From 2004 to 2006, he was a researcher at the Nagoya University Extension Course for Embedded Software Specialists.In 2006, he joined the Center for Embedded Computing Systems, Nagoya University, as an assistant professor, where he is now an associate professor.His research interests include system-level design automation and real-time operating systems.He received the best paper award from IPSJ in 2003. He is a member of ACM, IEEE, IPSJ, IEICE, and JSSST.



Ryo Kurachi is a Designated Associate Professor at Center for Embedded Computing Systems, Nagoya University. He graduated from Tokyo University of Science with undergraduate majors in applied electronics. After a few years working at AISIN AW CO., LTD. as a software engineer, he received his master's degree from Tokyo University of Science in 2007, followed by his Ph.D. in information science from the Nagoya University in 2012. His research interests includes embedded systems and real-time systems. Within that domain, he has investigated topics such as in-vehicle networks and real-time scheduling theory and embedded systems security. He is a member of IPSJ and IEICE.



Naoya Chujo received his B.E. degree in applied physics and his M.S. degree in information science and his Ph.D. degree in electrical engineering from Nagoya University in 1980, 1982 and 2004. He joined Toyota Central R&D Labs. in 1982. Since 2010, He is a professor at Aichi Institute of Technology. His research interests are in the area of embedded system and automotive electronics. He is a member of IEEE, the Information Processing Society of Japan (IPSJ), the Institute of Electronics, Information and Communication Engineers (IEICE),the Institute of Electrical Engineers of Japan (IEEJ), and Informatics Society.

Regular Paper

Classification Method of Unknown Web Sites Based on Distribution Information of Malicious IP addresses

Shihori Kanazawa[*], Yoshitaka Nakamura[**], Hiroshi Inamura[**], and Osamu Takahashi[**]

[*] Graduate School of Systems Information Science, Future University Hakodate, Japan
[**] School of Systems Information Science, Future University Hakodate, Japan
{g2116011, y-nakamr, inamura, osamu}@ fun.ac.jp

Abstract - In recent years, cyber-attacks via Web sites such as Drive-by download attacks or phishing attacks increase rapidly. The attackers acquire personal information of users illegally by these attacks and inflicts economical damage to the users. Therefore, it is important to detect malicious Web sites which cause economic damage. The conventional detection method of known malicious Web sites uses blacklists. And the detection method of unknown malicious Web sites uses the features of the domain name. Since it is relatively easy to change the domain name, it is inappropriate for the method of detecting malicious Web sites to using domain names. On the other hand, there is a characteristic that it is difficult to change the IP address once it is set. In this paper, we propose a method to classify Web sites as malicious or benign by using only a part of the network address, to reduce the classification cost. And, we analyzed features of the network address part of the IP address class to classify unknown Web sites. We evaluated our proposed classification method by cross-validation. As a result of evaluation, high classification accuracy was provided in IP address Class A. From this result, we could confirm the effectiveness of the proposed method. Also, as a result of changing the acquisition method of features of IP address based on the hypothesis that temporal change exists in the features of malicious IP address, we could confirm improvement of classification accuracy.

Keywords: cyber-attack, Drive-by download, malicious Web site, network address, IP address of Class

1 INTRODUCTION

In recent years, the threat of attacks by viruses or malwares on the Internet are increasing year by year. Among them, attacks via Web sites are increasing rapidly. According to "10 Major Security Threats 2017" [1] announced by Information-technology Promotion Agency, Japan (IPA), Attacks on Web sites appear in 1st, 4th, and 6th place in March 2017. As an example of attack, cyber-attacks via Web sites such as Drive-by download attacks or phishing attacks increase rapidly. Drive-by download attacks make users download malicious programs such as viruses or malwares. Phishing attacks navigate to fake Web sites. Attackers acquire personal information of users illegally by these attacks and inflicts economical damage. Figure 1 shows the number of incident occurrence and damage amount data from Ref. [2] by National Police Agency, Japan. In 2012,

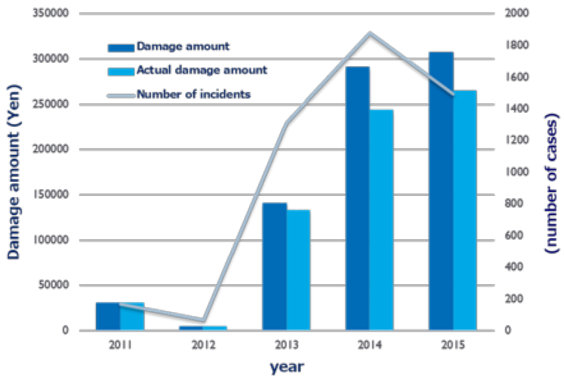


Figure 1: The number of incidents, damage amount, and actual damage amount

the total number of illegal acquisitions of personal information was only 50 cases for a year. However, the number of incident occurrence has increased to 1,400 cases for a year until 2015. In order to prevent such damage, it is necessary to take measures to prevent users from accessing malicious Web sites.

There are some methods to prevent users from accessing malicious Web sites. One method is to use Web reputation system [3]. The Web reputation system has a function to block malicious Web sites. If the connected domain name is judged to be malicious, the system blocks the access. In this way, the Web reputation system prevents damage caused by malicious programs or phishing. However, the Web reputation system can block only access to Web sites clearly made malicious activity such as virus distribution and phishing scams. Another method is to use the Intrusion Prevention System (IPS) [4]. IPS prevents sophisticated and advanced security threats such as bot attacks and DoS attacks that are difficult to protect only by general firewalls and anti-virus software. IPS examines the contents and behaviors of communication packets, and blocks Web access if IPS detects communication as malicious. There are two types of detection mechanisms in IPS. The first mechanism uses the signature, that is attack patterns collected in database(DB) in advance. If the access data in the communication matches the pattern in the DB, the data is regarded as malicious and the communication is blocked. The second is the anomaly detection mechanism. The mechanism uses a whitelist of benign operations. If the access is not in the whitelist, the anomaly detection mechanism regards the access as malicious and blocks the communication. Therefore, IPS can only detect known suspicious packets included in Web access communication.

Since the above two methods use known information such as information of suspicious packets included in known malicious Web sites, there is an advantage that the detection rate of known malicious Web sites is relatively high. However, these methods have drawbacks that cannot be detected unknown malicious Web sites. Therefore, it is also unknown whether these methods can obtain sufficient accuracy. In order to solve such problems, it is necessary to consider the detection conditions that enables to detect malicious Web sites including unknown Web sites. Also, it is necessary to classify unknown Web sites into benign Web sites and malicious Web sites. Therefore, we propose a method to detect and classify unknown malicious Web sites.

This paper is constructed as follows. Section 2 mentions about researches related to our study. Section 3 mentions the requirements of the proposed method. Section 4 mentions our plan of the experiment and experimental results. Finally, this paper concludes, and discusses on future work in Section 5.

2 RELATED WORK

In recent years, some system has been developed to prevent the user from accessing to malicious Web sites. For example, one of them is Web reputation system [3],[5] which detects malicious Web sites. This system uses a list of known malicious Web sites. The list of known malicious Web sites is called the blacklist. However, this system using the blacklist cannot deal with unknown malicious Web sites.

There are some other researches to detect malicious Web sites including researches based on the features of URLs, researches based on the features of domain names, and researches based on the features of IP addresses. Each type of research is described in detail below.

First, we introduce the researches based on the feature of URLs. Ma's group, and Tanaka's group proposed a supervised learning approaches for classifying URLs as malicious or benign based on the lexical structure of URLs respectively [6],[10]. These approaches can classify malicious domains and benign domains by the features that can be extracted from the DNS communication. Next, research is based on the feature of domain names. Ryu's group use the features of the domain names as the detection condition [7]. A malicious domain name has a feature that the length of the domain name is 10 or more characters and alphanumeric characters are mixed. The malicious domain name has these features because it is often generated automatically using Fast-Flux attack method [8]. Fast-Flux uses computers infected with bots (botnet) to distribute viruses or guidance information for phishing attacks. Bilge's group proposed a system that employs DNS analysis techniques to detect domains that are involved in malicious activity [9]-[10]. This system can classify malicious domains and benign domains by the features that can be extracted from the DNS communication. Bilge's group analyzed the DNS communication log over several months using these features and show that malicious domains can be detected with high accuracy [9]. These researches are effective for detection of known Web sites, because these researches use the blacklist of domain names and URLs. However, these researches cannot maintain the blacklist up to date easily, because domain names of Web site can be easily and continuously changed.

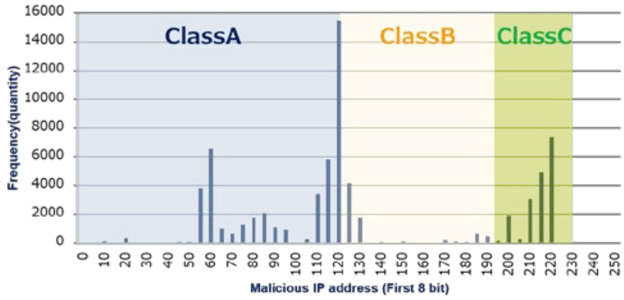


Figure 1: Usage distribution of IP addresses

As research based on IP addresses, Chiba's group proposed a method of utilizing the feature of malicious IP addresses [11]-[12]. This method classifies malicious IP addresses and benign IP addresses by the feature of malicious IP addresses, because cyber-attack is prone to use particular IP addresses [11]-[13]. This method has an advantage that the classification is difficult to be avoided, since publicly registered IP addresses cannot be easily changed. However, this method has two problems. First, classification range of IP addresses is narrow, because the IP addresses that can acquire features are limited. Second, classification cost is high, because high-dimensional feature set is required to represent the features of IP addresses. From these researches, it can be said that domain-based detection approaches using blacklists tend to fail. And it can be also said that it is difficult to use domain names to classify malicious Web sites since domain names can be easily changed. To solve these problems, we propose a new method to detect a malicious Web site effectively by reducing avoidance from the blacklist. This method uses only the domain name for detection to expand the detection range of the malicious Web sites. We also propose a method to classify Web sites by using a part of IP address to reduce the classification cost.

3 CLASSIFICATION METHOD OF MALICIOUS WEB SITES

3.1 Approach

In this research, we propose a system to classify malicious Web sites using domain name features and IP address features together. In the detection using the domain name feature, we solve the drawback of the blacklist type detection by extending the detection condition using multiple detection conditions used in existing methods together. In the classification using the features of the IP address, we aim to maintain high classification accuracy while reducing classification cost by acquiring the Web site’s distribution features in all ranges of the IP address and expressing each feature with small amount of information.

According to Ref. [14], the distribution of malicious IP addresses is as shown in Fig.2. In Ref. [11], authors described that cyber-attacks tend to use specific IP address

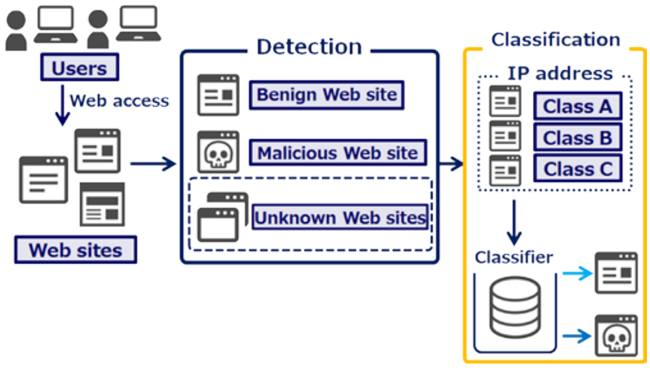


Figure 2: Approach to reduce the classification cost

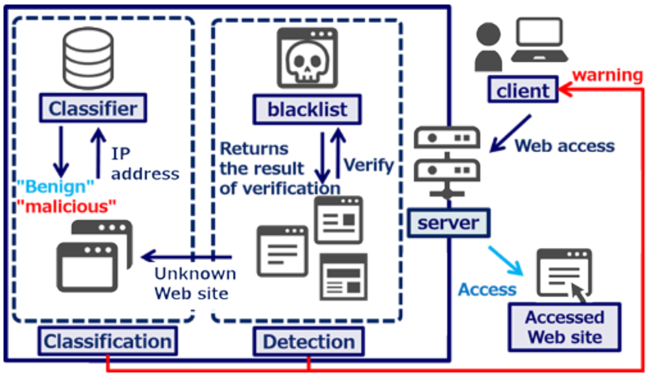


Figure 3: Overview of the proposed method

groups, and the usage frequency of malicious IP addresses also has different features for each IP address class. Using features of this usage frequency, we obtain the features of the range of all IP addresses and use it for classification. Also, based on the features of the network address space, only the network address part of each IP address is used for classification after binary conversion of the IP address. With this approach, it is possible to reduce the number of dimensions of feature vectors required for classification and enable high precision and low-cost classification.

3.2 Proposed Method

Figure 3 shows the outline of the proposed method using the approach to reduce classification cost. Web sites accessed by clients can be classified into three categories: malicious Web sites, benign Web sites, and unknown Web sites. It is important how to deal with these unknown Web sites. In this approach, we will classify unknown Web sites to malicious or benign at low cost by using IP addresses of only unknown Web sites.

Figure 4 shows the overview of the whole proposed method. The proposed method consists of two units. One is a unit to detect unknown Web sites by removing known malicious and benign Web sites. Another unit is to classify whether an unknown Web site is malicious or benign.

Details of the detection unit are described in Section 3.3, and details of the classification unit are described in Section 3.4.

The detection unit detects an unknown Web site by excluding malicious Web sites from the detection target using the blacklist of the domain name and transmits the IP address of the corresponding Web site to the classification unit.

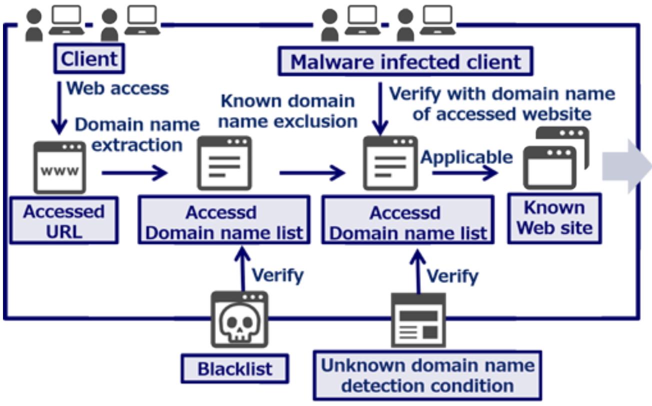


Figure 4: Details of detection method of unknown Web site

The classification unit classifies whether the unknown Web site is a malicious or benign Web site by using the features of the IP address transmitted from the detection unit. If the classification result is a benign Web site, the proposed method allows the client to access the Web site. On the other hand, if the classification result is a malicious Website, the proposed method interrupts communication by warning the client Also, the proposed method adds relevant malicious Web sites to the blacklist and keeps it up-to-date.

3.3 The Detection Method of Unknown Web Sites

Figure 5 shows details of the detection unit which detect unknown Web sites.

In the detection unit, unknown Web sites are detected using the features of the domain name. The domain name can be obtained from the URL of the Web site. First, when communication from a client to a certain Web site passes through the DNS server, the detection unit collates the domain name of the destination against the blacklist and excludes known malicious domains. If the domain name of the destination does not exist in the blacklist, the domain name is compared with the detection condition based on the features of the domain name such as the length of the domain name, character type, and so on. The Web sites that do not satisfy these conditions are defined as unknown Web sites and are determined as classification targets in the classification unit.

Also, the proposed method uses the domain name accessed by the clients infected with malware to check unknown Web sites. Figure 6 shows details of detection method of malware infected clients. Generally, malwares attempt clients to access numerous malicious Web sites to expand infection of malware. Therefore, malicious Web sites are likely to be accessed simultaneously from multiple malware infected clients [10]. Tanaka's group proposed the method to search clients accessing Web sites with malicious domain.

The clients detected by this method are called malware infected clients. By applying this method, it is possible to detect malicious Web sites with malicious domains that many clients are forced to access. In our proposed method, we use

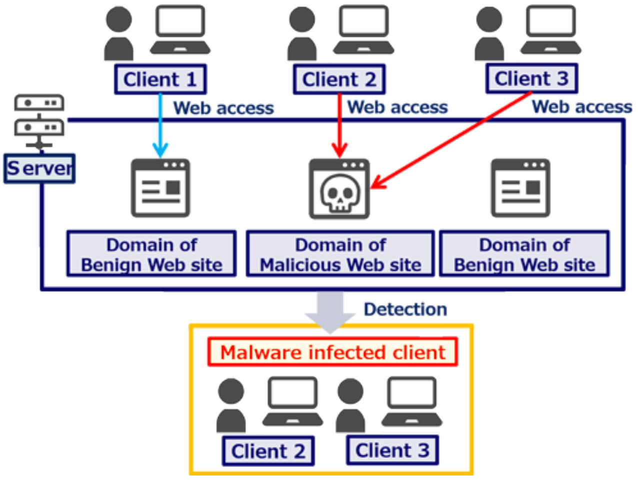
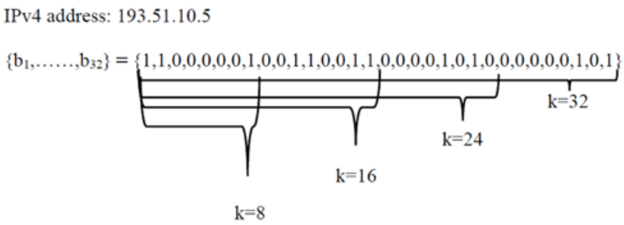


Figure 5: Detection method of malware infected clients



k	Feature vector
8	b_k=1 (k=1, 2, 8) b_k=0 (otherwise)
16	b_k=1 (k=1, 2, 8, 11, 12, 15, 16) b_k=0 (otherwise)
24	b_k=1 (k=1, 2, 8, 11, 12, 15, 16, 21, 23) b_k=0 (otherwise)
32	b_k=1 (k=1, 2, 8, 11, 12, 15, 16, 21, 23, 30, 32) b_k=0 (otherwise)

Figure 6: Examples of generating feature vectors

this malware infected clients to regard some Web sites as the known malicious Web sites. Finally, the IP address of Web sites which do not match the blacklist and the domain name detection conditions are used for classification and send to the classification unit.

3.4 The Classification Method of Malicious Web Sites

The classification unit is composed of two phases. First, the unit generates feature vectors from data of a blacklist in which known malicious Web sites are accumulated and data of a whitelist in which known benign Web sites are accumulated and uses these vectors as training data to construct classifiers (Section 3.4.1). Next, the unit classify unknown Web sites using the constructed classifiers (Section 3.4.2).

3.4.1 Construction of Classifier Using Training Datasets

In order to construct classifier, it is necessary to generate feature vectors. These feature vectors contain various features of the training datasets. Our method vectorizes the features of known malicious Web sites and benign Web sites and generate classifiers for classification using these feature vectors. The number of elements of the feature vector is called the dimension number. Since the number of dimensions of these feature vectors influences the cost of classification, we use the feature vectors with reduced dimension.

Table 1: Examples of labeling feature vectors of training datasets

IP address	Feature vector	Label
193.51.10.5	1,1,0,0,0,0,0,0,0,0,1,1,0,0,1,1	1
10.10.10.10	0,0,0,0,1,0,1,0	1
203.4.12.89	1,1,0,0,1,0,1,1,0,0,0,0,0,1,0,0,0,0 ,0,1,1,0,0	0
…	…	…

Figure 7 shows a feature vector generation method. First, the target IP address is converted into bit string. Next, all bit strings are represented as k-dimensional vector $\{b_1,......,b_k\}$. 3 types of feature vectors are generated according to the IP address class such as Class A, Class B, and Class C. A vector of 8 dimensions in the case of Class A, a vector of 16 dimensions in the case of Class B, and a vector of 24 dimensions in the case of Class C are generated for each IP address. Finally, generated feature vectors of malicious are labeled as “1”, and generated feature vectors of benign are labeled as “0” in this system. Table 1 shows examples of labeling the feature vector of the training data set. The classifier used in this method is Support Vector Machine (SVM), which is one of pattern identification methods. Since we examined the method with the goal of improving classification accuracy for related research, we use SVM to keep the same condition. In addition, Ref. [6] clearly indicated that malicious Web sites are detected with high accuracy by using SVM. In addition, when the features of the IP addresses are frequently updated, an approach such as on-line learning may be suitable. Online SVM is one of such identification method. Since it is possible to follow the time change of learning data, on-line learning approach may be effective in the proposed method which learns data change on time. However, since the observational data set we are using is released every year, the frequency of re-learning by updating the data is presumed to be several times a year at most, as long as the data update frequency is this level. Therefore, we judged that batch type SVM algorithm has sufficient performance at present, and the proposed method classifies using the batch type SVM.

The proposed method constructs three classifiers based on feature vectors described above.

3.4.2 Classification of Test Datasets Using Constructed Classifier

An unknown IP address sent from the detection unit are classified by the constructed classifiers described in Section 3.4.1. Figure 8 shows that classification has three steps. First, the IP address passed from the detection unit is classified for

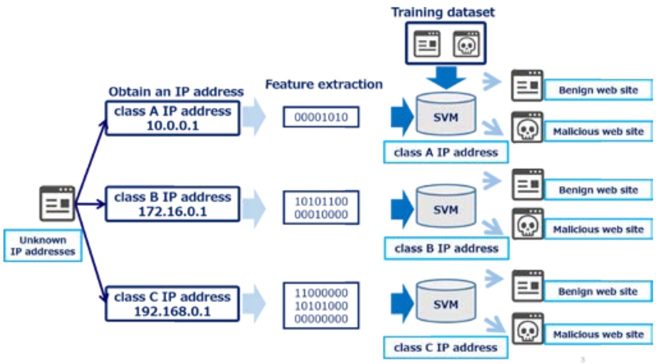


Figure 7: Process of the classification

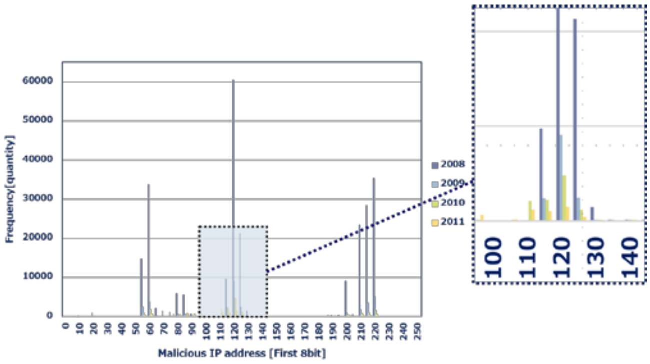


Figure 8: Usage frequency of malicious IP addresses (Enlargement of the first 8 bits of IP address around 120) (2008-2011)

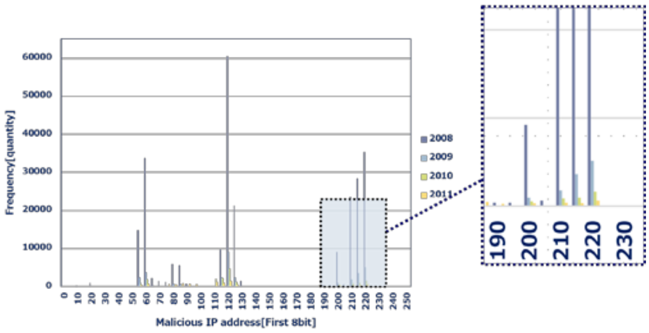


Figure 9: Usage frequency of malicious IP addresses (Enlargement of the first 8 bits of IP address 190-230) (2008-2011)

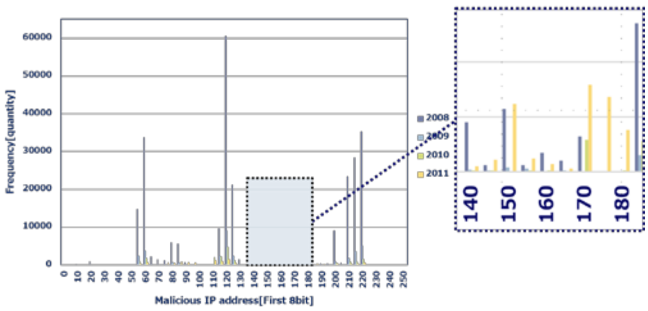


Figure 10: Usage frequency of malicious IP addresses (Enlargement of the first 8 bits of IP address 140-180) (2008-2011)

each IP address class in the classification unit. Next, feature vector is generated from this obtained unknown IP address. This feature vector is classified as malicious or benign by the classifiers.

Then, the classifier is updated by using IP addresses already known as malicious or benign and IP addresses for which classification has been completed as training data. As a result, the features of the malicious IP address distribution included in the classifier can maintain the latest state, and the classification accuracy can also be improved for unknown Web sites.

In order to realize these proposed methods, it is necessary to set the dataset used as training data in a state suitable for classification. Therefore, we analyzed the usage distribution of IP addresses class based on the data on malicious IP address usage frequency. The classifier pattern to be considered from the analysis result is the following three.

1) Classifiers using blacklist itself (General blacklist)
2) Classifiers using the features of IP addresses on blacklist without assuming temporal change.
3) Classifiers re-learned the temporal changes on the features of IP addresses on blacklist

In case of 3), it is necessary to analyze the IP address used as the training data to confirm whether it can cope with the temporal changes of the IP address included in the blacklist.

3.5 Temporal Change on IP address Distribution

We analyzed the usage status of each IP address class to check whether the distribution of malicious IP addresses changes over time. Figure 9 shows the usage frequency of malicious IP addresses from 2008 to 2011, when the number of IP addresses collected from CCC datasets [15] which is a bot observation data group containing malware is relatively large. At the part where the first 8 bits of the IP address are around 120, we can see that the usage frequency has decreased from 2008 to 2011. Also, Fig.9 shows at the part where the first 8 bits of the IP address are around 110.The malicious IP address is not used from 2008 to 2009. However, we can see that the usage frequency has increased from 2010. Also, Figure 10 shows at the part where the first 8 bits of the IP address are from 200 to 220. We can see that the usage frequency has decreased from 2008 to 2011.

In addition, Fig. 11 shows at the part where the first 8 bits of the IP address are from 170 to 180. We can see that the usage frequency has increased from 2010 to 2011. Therefore, it can be said that the temporal change of IP address distribution features. In addition, Fig.12 shows at the part where the first 8 bits of the IP address are from 170 to 180. we can see that the usage frequency has increased from 2010 to 2011. It can be said that the temporal change of IP address distribution features. From this result, the distribution of malicious IP addresses has changed every year in each address class. By extracting the features of each IP address class by each year and using the feature for classification, highly accurate classification corresponding to temporal changes in IP address usage can be realized. Using these features, the training data used for constructing the SVM classifier in the classification unit is changed every year in the proposed system.

4 EVALUATION

4.1 Outline

In order to confirm the effectiveness of the proposed method, we evaluated the classifier constructed in phase 1 of the classification unit by three evaluation items such as accuracy, precision rate, and recall rate. In this paper, we define True Positive (TP) as a number that can classify malicious IP address correctly as malicious IP address, and True Negative (TN) as a number that can classify benign IP address correctly as benign IP address. And we also define False Positive (FP) as a number that can classify benign IP address incorrectly as malicious IP address, and False Negative (FN) as a number that can classify malicious IP address incorrectly as benign IP address. Accuracy (1), precision rate (2), recall rate (3) are calculated by the following formulas.

$$Accuracy = (TP + TN) \,/\, (TP + TN + FP + FN) \tag{1}$$

$$Precision = TP \,/\, (TP + FP) \tag{2}$$

$$Recall = TP \,/\, (TP + FN) \tag{3}$$

The dataset of benign and malicious are obtained from Malware Workshop (MWS) Datasets [14]. As malicious data, we use the CCC dataset (2008-2011) which is a bot observation data group containing malware specimens, and D3M (2010-2015) which is Web infectious malware data, from MWS dataset. As benign data, we use 50,000 data of Alexa Top Global Sites [15] (2016) and data created based on NCD in MWS Cup (2014) which is a white dataset. For the experiments, we extract the ratio of malicious IP address to benign IP address to be 8: 2, 5: 5, 2: 8 respectively from these datasets. Training datasets and test datasets are randomly extracted from malicious and benign data respectively.

In the previous section we confirmed the temporal change of the distribution of the malicious IP addresses. In order to evaluate the influence of this temporal change on classification accuracy, we compare the accuracy of the three types classifiers by using the first 8, 16, 24, 32 bits of each IP address.

4.2 Evaluation Experiment 1

In evaluation experiment 1, we compare the classification using blacklist and the classification of the proposed method to confirm the classification performance of each classifiers. We created 5 types of blacklists using malicious IP addresses that can be obtained from the datasets [14] for classification using blacklists. Each blacklist consists of 17000 IP addresses. We also created 5 types of test datasets for classification of the proposed method using malicious IP addresses and benign IP addresses that can be obtained from the datasets [14]. This test datasets consists of 20000 IP addresses, and the ratio of malicious IP address to benign IP address is 8:2 in each test datasets. The overview of evaluation experiment 1 is shown in Fig.12. We conducted experiments to compare the results of classification of two patterns using the same input test datasets. Pattern 1 is classification using blacklist, and pattern 2 is classification using the proposed method. In experiment pattern 1, 32-bit IP address is used as blacklist data. The classifier classifies by comparing the IP address of the input data with the IP address of the blacklist, and outputs the classification result. We calculate the average classification accuracy of 5 input test datasets. For experiment pattern 2, we use the same dataset as experiment pattern 1.

We construct classifiers of the proposed method using the dataset as training data. When test dataset is input, the classifiers of the proposed method classify according to the IP address class of the input IP address and outputs the classification result. We calculate the average classification accuracy by 5-fold cross validation. Table 2 shows the number of IP addresses used in evaluation experiment 1.

Table 3 shows the classification results of classification of the experiment pattern 1. On average, the accuracy was 18.762%, the precision rate was 5.268%, the recall rate was 100%, and accuracy and precision rate are very low.

Table 4 shows the average classification result of the experiment pattern 1. From this result, accuracies and precision rates were more than 80%, and recall rates were more than 90%. Compared to the classification result using blacklist, high values were obtained.

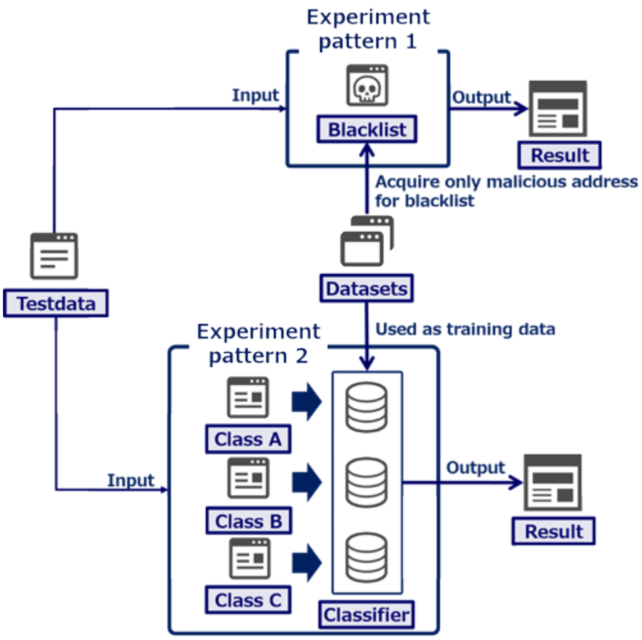


Figure 11: Overview of evaluation experiment 1

Table 2: Number of IP addresses used in experiment 1

	Malicious IP addresses	Benign IP addresses
Blacklist	322,687	
Whitelist		538,156

Table 3: Detection result using blacklist

	Accuracy	Precision	Recall
Testdata1	18.500	5.111	100
Testdata2	18.545	5.185	100
Testdata3	18.605	5.305	100
Testdata4	19.335	5.372	100
Testdata5	18.825	5.368	100
Average	18.762	5.268	100

Table 4: Average of classification results of the proposed method

	Accuracy	Precision	Recall
Testdata1	81.945	86.543	91.983
Testdata2	81.670	86.385	91.774
Testdata3	81.520	86.177	91.747
Testdata4	81.782	86.743	91.744
Testdata5	81.735	86.442	91.751
Average	81.730	86.458	91.800

Table 5: Number of IP addresses used in experiment 2

	Malicious IP addresses	Benign IP addresses
IP addresses of Class A	49,164	40,667
IP addresses of Class B	3,523	10,735
IP addresses of Class C	75,000	14,288

Table 6: Result of IP address of Class A

	Accuracy	Precision	Recall
Case1(k=8)	84.060	90.253	89.866
Case2(k=16)	83.743	89.893	89.797
Case3(k=24)	83.760	89.890	89.818
Case4(k=32)	83.878	89.887	89.953

Table 7:Result of IP address of Class B

	Accuracy	Precision	Recall
Case1(k=8)	83.541	92.157	87.855
Case2(k=16)	81.696	89.070	88.162
Case3(k=24)	82.945	90.276	88.610
Case4(k=32)	83.257	90.525	88.761

Table 8: Result of IP address of Class C (8:2)

	Accuracy	Precision	Recall
Case1(k=8)	81.781	90.524	87.191
Case2(k=16)	81.201	89.965	86.981
Case3(k=24)	81.243	90.000	87.001
Case4(k=32)	81.222	90.586	86.564

Table 9: Result of IP address of Class C (5:5)

	Accuracy	Precision	Recall
Case1(k=8)	67.252	73.824	65.248
Case2(k=16)	67.252	73.824	65.248
Case3(k=24)	67.280	73.712	65.310
Case4(k=32)	66.034	69.050	65.122

4.3 Evaluation Experiment 2

In evaluation experiment 2, we experimented in detail to examine whether classification into IP address class is appropriate. We confirm the effectiveness of the classification using features of each IP address class. For each IP address corresponding to the IP address classes A, B, and C, we evaluated the classification accuracies using the first 8 bits, 16 bits, and 24 bits part of the IP address which is the network address part of each address class as feature.

The evaluation procedure is as follows. First, test datasets are input to the proposed method with the classifier generated by the training data, and classification result is acquired. Next, the classification accuracy is calculated by 5-fold cross validation. Finally, the classification accuracy for each feature vector used for input is compared and evaluated. The number of IP addresses used in evaluation experiment 2 is shown in Tables 5.

Evaluation experiment 2 is conducted in 4 cases. For the number of first bits of the IP address used for classification, the classification results were evaluated by 8 bits for Case 1, 16 bits for Case 2, 24 bits for Case 3, and 32 bits for Case 4.

Table 6 shows the classification results for IP address class A. The accuracy and the precision rate achieved the highest value in Case 1 using only the network address part of IP address class A as the feature vector. On the other hand, the recall rate achieved the highest value in Case 4. Although the accuracies of the Ref. [11] were from 74.7% to 75.1%, the accuracy of the proposed method was 84%.

Table 7 shows the classification results for IP address class B. The accuracy and the precision rate achieved the highest value in Case 1, and the recall rate achieved the highest value in Case 4. Therefore, in Case 2 where only the network address part of IP address class B is used as the feature vector, the highest value cannot be achieved. The accuracies of the proposed method were 81.6% to 83.5% which were lower than the accuracies of the Ref. [11] (84.6% to 86.2%).

Table 8 shows the classification results for IP address class C. The accuracy and the recall rate achieved the highest value in Case 1, and the precision rate achieved the highest value in Case 4. The accuracy, the recall rate, and the precision rate of Case 3 where only the network address part of IP address class C are used as the feature vector were lower than those of Case 1, 2 and 4. Also the accuracies of the proposed method were 81.2% to 81.7% which were lower than the accuracies of the Ref. [11] (85.1% to 88.5%).

Table 9 shows the classification results when the ratio of malicious IP address to benign IP address is set to 5: 5 in IP address class C. The accuracy and recall rate achieved the highest value in Case 3 using only the network address part of IP address class C as the feature vector. The precision rate achieved the highest value in Case 1 and 2. Also, in Case 4, the accuracy, the precision rate and the recall rate achieved the lowest values.

4.4 Evaluation Experiment 3

In evaluation experiment 3, we made classification based on training data using features of IP address every year. And we evaluate the effectiveness of the training data in each year according to the accuracy obtained every year. The number of IP addresses used in evaluation experiment 3 is shown in Tables 10, 11, and 12.

Figure 13 shows the results with IP addresses corresponding to IP address class A. The accuracy of classification varies depending on the year.

Figure 14 shows the results with IP addresses corresponding to IP address class B. While the classification accuracy of Ref. [11] is from 84.6% to 86.2%, the classification accuracy of the proposed method exceeded 90% from 2008 to 2009.

Figure 15 shows the results with IP addresses corresponding to IP address class C. While the classification accuracy of Ref. [11] is from 85.1% to 88.5%, the classification accuracy of the proposed method exceeded 85% in 2008 and 2011.

Table 10: Number of IP addresses of Class A used in experiment 3

	Malicious IP addresses	Benign IP addresses
2008	159,103	40,897
2009	22,537	40,897
2010	14,369	40,897
2011	9,538	40,897

Table 11: Number of IP addresses of Class B used in experiment 3

	Malicious IP addresses	Benign IP addresses
2008	200	10,735
2009	22,537	10,735
2010	14,369	10,735
2011	945	10,735

Table 12: Number of IP addresses of Class C used in experiment 3

	Malicious IP addresses	Benign IP addresses
2008	97,688	14,288
2009	11,619	14,288
2010	4,073	14,288
2011	1,854	14,288

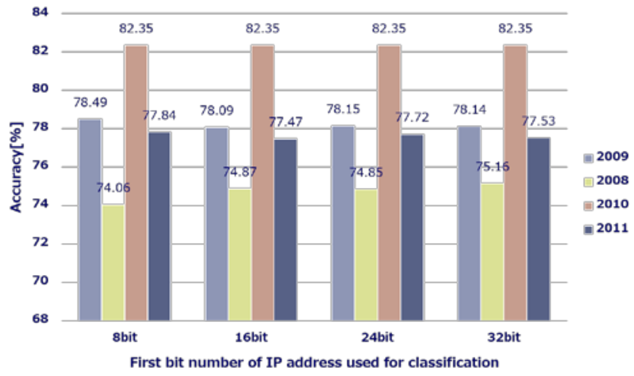


Figure 12: The result of evaluation experiment 3 in IP address of Class A

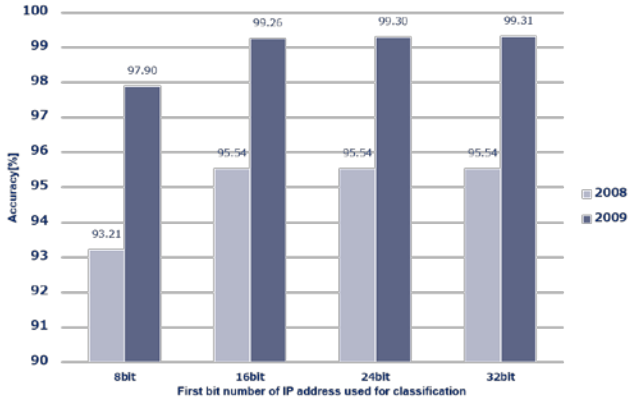


Figure 13: The result of evaluation experiment 3 in IP address of Class B

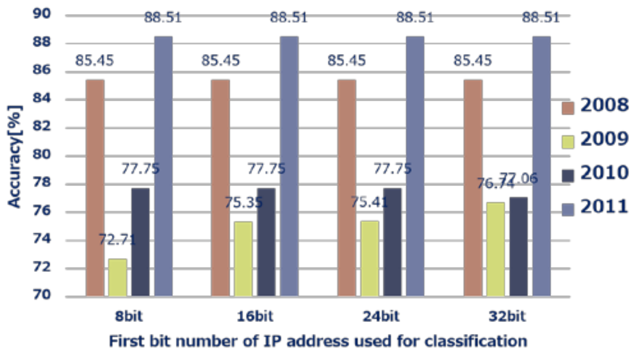


Figure 14: The result of evaluation experiment 3 in IP address of Class C

4.5 Discussion

From the results of evaluation experiment 1, since the accuracy showed low values, the classification of malicious Web sites using blacklist has low generalization ability and has no ability to deal with unknown Web sites. From the result of evaluation experiment 2, the proposed method is effective for unknown malicious Web sites with the address of IP address class A, because the sufficient accuracy can be maintained by classification using only the network address part. On the other hand, as for the addresses of IP address class B and IP address class C, overall accuracy is lower than those of IP address class A. There are two possible causes. First, since the number of IP addresses of the training data set is extremely small, there is a possibility that the feature may not clearly appear. Second, there is a possibility that time changes may occur in the features of malicious IP addresses. Finally, we describe the experimental results of evaluation experiment 3. From section 3.6, it was found that time changes were observed in the features of malicious IP addresses.

Based on these analysis, it is possible to improve classification accuracy by considering temporal change of features of malicious IP addresses. In Ref. [11], the classifiers are constructed with all bits of IP address as features. However, in the proposed method, it is shown that the same accuracy can be obtained by using the features of the temporal change of the appearing IP address distribution even with the classifiers using only the network part of each IP address. From these results, it is difficult to maintain accuracy by the method that only accumulates and uses the malicious IP address data group (blacklist). However, by changing the state of training data for each IP address class and each year, it is possible to maintain the features of malicious IP address and to maintain classification accuracy. Therefore, it is possible to deal with unknown Web sites by finding the range of network address group with variation of features among IP address classes.

5 CONCLUSION

In this paper, we proposed a method of detecting unknown Web sites and classifying Web sites as benign or malicious by using only a part of the network address, in order to reduce the cost of the classification. As a result of evaluation experiment about classifiers using the features of IP addresses on blacklist without assuming temporal change, high classification accuracy was provided in IP address Class A, and we confirm the effectiveness of the proposed classification method. However, from the results of evaluation experiment about classifiers re-learned the temporal changes on the features of IP addresses, it is shown that the same accuracy can be obtained by using the features of the temporal changes of the appearing IP address distribution even with the classifiers using only the network part of each IP address.

In the future, further research should be done to investigate the feature of malicious Web sites to improve the accuracy of classification. For more effective classification, further analysis of the data in each IP address class. On the other hand, although significant difference in features is observed for each IP address class in the current usage status of IP address, considering the address distribution by CIDR, there is a possibility of further improving the classification accuracy. However, although there are many CIDR IP addresses by 24-bit prefix corresponding to IP address class C in the investigation result by Ref. [16], the influence on the classification accuracy is unknown in the case of using addresses with other prefix lengths. Also in the classification method using the IP address, it is necessary to study how to classify it as a benign IP address when a new benign Web site is constructed for the IP address once classified to be malicious. In addition, since we examined the method with the goal of improving discrimination accuracy for related research, we use SVM to keep the same condition. In addition, although the proposed method uses SVM according to related research at present, in the future it is necessary to examine concrete classifier selection and design.

REFERENCES

[1] Information-technology Promotion Agency, Japan, "10 Major Security Threats" <https://www.ipa.go.jp/files/000058504.pdf> [Accessed May 31, 2017] (*in Japanese*).

[2] National Police Agency, "Public information of the National Police Agency: About the occurrence situation of illegal remittance offenses related to Internet banking in Heisei 26, " <https://www.npa.go.jp/cyber/pdf/H270212_banking.pdf> [Accessed January 17, 2017] (*in Japanese*).

[3] R. Farmer, and B. Glass, "Building Web Reputation Systems, " Yahoo! Press, (2010).

[4] "What is IDS/IPS? | JUNIPER NETWORKS," <https://www.juniper.net/us/en/products-services/what-is/ids-ips/> [Accessed Oct 19, 2017]

[5] M. A. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt, "Trends in circumventing web-malware detection," Google, Google Technical Report, (2011).

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious Web sites from suspicious urls," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'09), pp. 1245–1254, (2009).

[7] I. Ryu, "Detection method of malicious site by DNS information," Master's thesis of Waseda University (2012).

[8] Hitachi Solutions, Ltd. "Information security blog," <http://securityblog.jp/words/2898.html> [Accessed January 17, 2017] *(in Japanese)*.

[9] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "Exposure Finding Malicious Domains Using Passive DNS Analysis," Proceedings of the 18th Annual Network & Distributed System Security Symposium (NDSS Symposium 2011), (2011).

[10] K. Tanaka, A. Nagao, and M. Morii, "Extracting Malicious Web site from DNS Log-Analysis Method and Anonymity-," Proceedings of the Computer Security Symposium 2013(CSS2013), pp.132-138, (2013) *(in Japanese)*.

[11] D. Chiba, K. Tobe, T. Mori, and S. Goto, "Analyzing Spatial Structure of IP Addresses for Detecting Malicious Web sites," Journal of Information Processing Vol.21, No.3, pp.539-550, (2013).

[12] D. Chiba, T. Mori, and S. Goto, "Deciding priority crawling in searching for malicious Web sites," Proceedings of the Computer Security Symposium 2012(CSS2012), pp.805-812, (2012) *(in Japanese).*

[13] D. Chiba, T. Yagi, M. Akiyama, and T. Mori, "Correlation Analysis Between IP Addresses Used in Varie-

ty of Attacks," Proceedings of the Computer Security Symposium 2011(CSS2011), pp.185-190, (2013) *(in Japanese)*.

[14] Y. Takata, M. Terada, J. Murakami, T. Kasama, K. Yoshioka, and M. Hatada, "Datasets for Anti-Malware Research ~MWS Datasets 2016~," IPSJ SIG Technical Report, Vol.2016-CSEC-74, No.17, pp. 1-8, (2016). (*in Japanese*).

[15] Alexa Internet, Inc., "The top 500 sites on the web," <http://www.alexa.com/topsites> [Accessed January 17, 2017].

[16] University of Oregon Route Views Project, <http://www.routeviews.org/> [Accessed December 27, 2017].

(Received October 20, 2017)
(Revised December 27, 2017)

Shihori Kanazawa received B.E. degree in systems information science from Future University Hakodate, Japan in 2016. She is a graduate student of Future University Hakodate. Her research interests include Web security using the detection of malicious Web sites. She is a student member of IPSJ.

Yoshitaka Nakamura received B.E., M.S., and Ph.D. degrees from Osaka University in 2002, 2004 and 2007, respectively. He is currently an associate professor at the School of Systems Information Science, Future University Hakodate. His research interest includes information security and ubiquitous computing. He is a member of IEEE, IEICE, and IPSJ.

Hiroshi Inamura He is a professor of School of Systems Information Science, Future University Hakodate, since 2016. His current research interests include mobile computing, system software for smart devices, mobile/sensor network, and their security. He was an executive research engineer in NTT docomo, Inc. He received B.E., M.E. and D.E. degree in Keio University, Japan. He is a member of IPSJ, IEICE, ACM and IEEE.

Osamu Takahashi received M.E. degree from Hokkaido University in 1975. He is currently a professor at the School of Systems Information Science, Future University Hakodate. His research interest includes ad-hoc network, network security, and mobile computing. He is a member of IEEE, IEICE.

Submission Guidance

About IJIS

International Journal of Informatics Society (ISSN 1883-4566) is published in one volume of three issues a year. One should be a member of Informatics Society for the submission of the article at least. A submission article is reviewed at least two reviewer. The online version of the journal is available at the following site: http://www.infsoc.org.

Aims and Scope of Informatics Society

The evolution of informatics heralds a new information society. It provides more convenience to our life. Informatics and technologies have been integrated by various fields. For example, mathematics, linguistics, logics, engineering, and new fields will join it. Especially, we are continuing to maintain an awareness of informatics and communication convergence. Informatics Society is the organization that tries to develop informatics and technologies with this convergence. International Journal of Informatics Society (IJIS) is the journal of Informatics Society.

Areas of interest include, but are not limited to:

Internet of Things (IoT)
Smart Cities, Communities, and Spaces
Big Data, Artificial Intelligence, and Data Science
Network Systems and Protocols
Computer Supported Cooperative Work and Groupware
Security and Privacy in Information Systems
Intelligent Transportation System
Distributed Computing
Multi-media communication
Information systems
Mobile computing
Ubiquitous computing

Instruction to Authors

For detailed instructions please refer to the Authors Corner on our Web site, http://www.infsoc.org/.

Submission of manuscripts: There is no limitation of page count as full papers, each of which will be subject to a full review process. An electronic, PDF-based submission of papers is mandatory. Download and use the LaTeX2e or Microsoft Word sample IJIS formats.

http://www.infsoc.org/IJIS-Format.pdf

LaTeX2e

LaTeX2e files (ZIP) http://www.infsoc.org/template_IJIS.zip

Microsoft Word™

Sample document http://www.infsoc.org/sample_IJIS.doc

Please send the PDF file of your paper to secretariat@infsoc.org with the following information:

Title, Author: Name (Affiliation), Name (Affiliation), Corresponding Author. Address, Tel, Fax, E-mail:

Copyright

Publisher

Address: Informatics Laboratory, 3-41 Tsujimachi, Kitaku, Nagoya 462-0032, Japan
E-mail: secretariat@infsoc.org

CONTENTS