Chen, W. et al. (Eds.) (2017). Proceedings of the 25th International Conference on Computers in Education.
New Zealand: Asia-Pacific Society for Computers in Education

System Architecture and Predictive Experiment for an Automatic Learning Support Function on Classroom Response Systems

Kozo Mizutani*
Department of Information and Electronic Engineering, School of Science and Engineering, Teikyo University, Japan.
* mizutani@ics.teikyo-u.ac.jp

Abstract: The Classroom Response System (CRS) was developed as a tool for enhancing interactivity between teachers and students in the classroom. Many studies have revealed the effectiveness of using CRS in educational environments. However, the popularity of CRS has not increased sufficiently. This study aims to implement an automatic learning support function on CRS to improve conventional systems. In this regard, I considered applying the concept of an agent-oriented system (AOS) to realize the implementation of this support function on web-based CRS. In addition, I examined the feasibility of applying the concept of an automatic learning support function by examining the prediction of student responses by using student activity logs. This paper describes the model on which the application of an AOS is based and the result of the prediction examination.

Keywords: Classroom Response System, Clicker, Learning Analytics, Multi-Agent System

1. Introduction

The Classroom Response System (CRS), also known as a Student Response System (SRS), Audience Response System (ARS), or Clicker, is a tool for enhancing interactivity between teachers and students in the classroom. The use of CRS has become accepted as a pedagogical tool capable of increasing the effectiveness of education. However, although the educational effect is appreciated, CRS has not become sufficiently popular. One of the reasons is considered to be that learning support functions that adapt to each student are not implemented on CRS or that its effect is insufficient. Therefore, my approach has involved considering methods to realize a learning support function that enables automatic and real-time improvement of further education using CRS. This paper describes the system architecture for realizing the automatic learning support function and the results of the prediction examination of student responses using activity logs based on the architecture.

2. Background

2.1 Related Work

Studies of the system architecture of CRS and the pedagogy of using the system in practice are important themes of information technology in education and have been attracting attention for the past few decades (Fujita et al. 1969). Recently, Karakostas et al. developed a CRS named "QuizIt" for Android devices (Karakostas et al. 2014). QuizIt consists of three parts: a mobile application for the lecturer, a mobile application for the students, and a web environment for lecturer administration. All of these components run in real time. The results of a pilot study showed that students were quite positive about the use of QuizIt in the laboratory course. Barth-Cohen et al. investigated the effect of promoting productive peer discussion among middle school science students using Clicker (Barth-Cohen et al. 2016). Their study recorded peer conversations to characterize the nature of student discourses and an analysis of these conversations. In another study using Clicker to investigate peer discussion, Majumdar et al. tried to visualize and categorize Clicker responses using the interactive stratified attribute tracking diagram (iSAT) (Majumdar et al. 2015). They concluded that iSAT contributes to conducting cohort analysis of teaching learning practice for teachers. The result revealed that students improve their performance on Clicker questions when they engage in peer discussion. Premkumar used Clicker in summative exams (Premkumar 2016). The results showed the process of an instructor is less time consuming, efficient, and more secure compared to the use of scan sheets. Furthermore, students were accepting the use of this technology in high stakes exams, and found it engaging and satisfying, primarily because it provides them with instant feedback. In addition, many studies about using CRS relating to the effectiveness of feedback and timing (Lantz et al. 2014), the effect of different teaching strategies (Liu et al. 2016), and so on, have been reported.

In addition to these studies about the use of CRS, other studies relating to evaluating the impact of using the CRS have also been published. For example, Hunsu et al. examined the potential effects of using CRS by comparing classrooms that did and did not use it to determine the different cognitive and non-cognitive learning outcomes (Hunsu et al. 2016). They discovered that CRS has a small but significant effect on cognitive learning outcomes and a near medium effect on non-cognitive learning outcomes. They also pointed out that instructors need to provide feedback in a constructive and timely manner. Richardson et al. mentioned the necessity of a standardized instrument to evaluate the impact of using the CRS, and developed an instrument named the CRiSP questionnaire (Richardson 2015). The questionnaire consists of 26 base items and includes three scales: the usability, the impact of using the CRS on engagement, and the impact of using the CRS on learning.

In summary, the aforementioned previous studies evaluating the impact of using CRS show that, although the educational effect is appreciated, CRS has not become sufficiently popular. Many studies have investigated the educational effect of using CRS; however, reducing the operational costs associated with CRS is important for its popularization. CRiSP does not include items of system operation. Moreover, we need even more discussion about the realization of automatic learning functions in which the system grasps the state of student understanding, presents additional teaching materials, or warns irresponsible students. Therefore, the learning effect of using CRS is expected to be increased by the introduction of automatic learning support functions.

2.2 Approach of This Study

Based on the above-mentioned background, this study developed two different CRSs named "Response Analyst" that runs on PCs and "WebCRS" that runs on smart devices. These CRSs have been used in actual classes (Mizutani 2013, 2014) and aim to realize automatic learning support functions by applying the experience that was gained during past use of these systems.

These systems have the following characteristics:

- Real-time communication:
  Because CRS helps communication in classes, real time is important. Response Analyst uses the original protocols to optimize the real-time response of the system. WebCRS relies on the use of WebSocket, which is a new web technology for real-time communication.
- Collect and Record Usage Logs:
  The systems have a function that collects and records usage logs in the form of an "Activity Log" with millisecond precision. On the students' devices, WebCRS not only makes it possible to transmit data relating to the basic functions such as receiving question contents, receiving a signal to start answering, and sending an answer, but also enables data to be collected by the accelerator sensor or inclination sensor of a smart device (Mizutani 2016). These data are recorded to a database (DB) in the background with the aim of reducing the influence on the basic functions of CRS by maintaining a small system load. In addition, the log data of the basic functions are collected with a timestamp of both the servers and each student's device. The timestamp of a student's device can be used to accurately analyze the state of a student. For example, the time that passes from the moment a student receives a start answering signal until the moment they submit their answer cannot be accurately analyzed by using the server time because of the time error caused by network delay and so on. By using a device timestamp, an accurate analysis of the difference of the time of each event is possible.

Moreover, WebCRS has the following characteristics, designed to improve the problems mentioned above in section 2.1.

- Using smart devices :
  WebCRS supports common web browsers pre-installed on PCs, smartphones, tablets, and so on.
- No installation required :
  To be able to start using WebCRS easily, WebCRS is implemented as a web application. If a web browser is installed on a device, we can use WebCRS without the installation of other programs. In case communication by WebSocket cannot be established by limitation of the network, browser version, and so on, WebCRS uses the conventional HTTP-based protocol. As result, using WebCRS as a “bring your own device” (BYOD) is possible too.
- Reducing Costs :
  The costs of using the entire system from preparation to maintenance after classes are reduced. By adopting BYOD, the cost of managing the student's device can be avoided. The Web servers, as the system core of WebCRS, also run on public cloud services.
- Scalability :
  The architecture of WebCRS is designed for cloud services; thus, it is easy to multiplex web servers. The design assumes simultaneous use at a scale ranging from one classroom to an entire school.

Figure 1 shows an example of a screenshot of the WebCRS interface. The functions for students and teachers are provided as web applications.

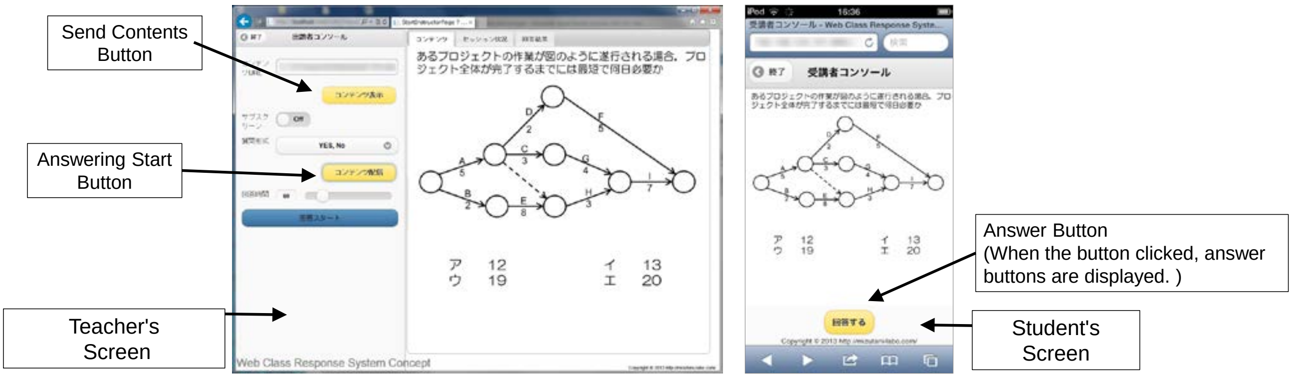

Figure 1. Examples of WebCRS screens.

The next section describes concrete methods intended to achieve the implementation of an automatic learning support function on WebCRS.

3. System Design for an Automatic Learning Support Function

3.1 An Automatic Learning Support Function

This study aims to realize an automatic learning support function as follows by using the activity logs of students on WebCRS.

- Providing supplementary teaching materials :
  To help a student to answer a question, the system decides whether to provide the student with supplementary teaching materials, and automatically provides these materials if the system considers this necessary.
- Warning to irresponsible students :
  The system detects irresponsible students who answer a question without sufficient thinking, and warns the students to learn positively.

- Notification to a teacher :
  The system notifies a teacher of the state of these supports.

In this section, concrete methods to implement these functions on WebCRS are considered.

3.2 Problems Associated with Implementing the Learning Support Function

The architecture of WebCRS, which is designed to enhance scalability, is shown in Figure 2. The system includes a conventional web application, web servers, and DB servers, but a load balancer (LB) is introduced to distribute the requests for user connections.

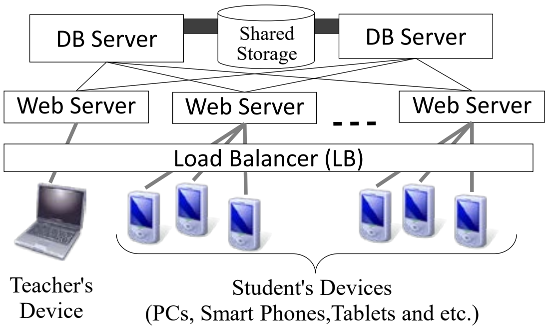

Figure 2. System structure of WebCRS.

The web server includes a mechanism implemented to process distributed requests by the LB. For example, as shown in Figure 3 (a), the request of Student A is connected to Node A by the LB. Node A stores the state of the connection as Session Info to the DB. Thereafter, the load balance of the system is changed, and the destination of the connection of Student A is changed to Node B by the LB, as shown in Figure 3 (b). In this case, because Node B does not have the Session Info of Student A, Node B obtains this information from the DB and maintains the state of the connection continuously.

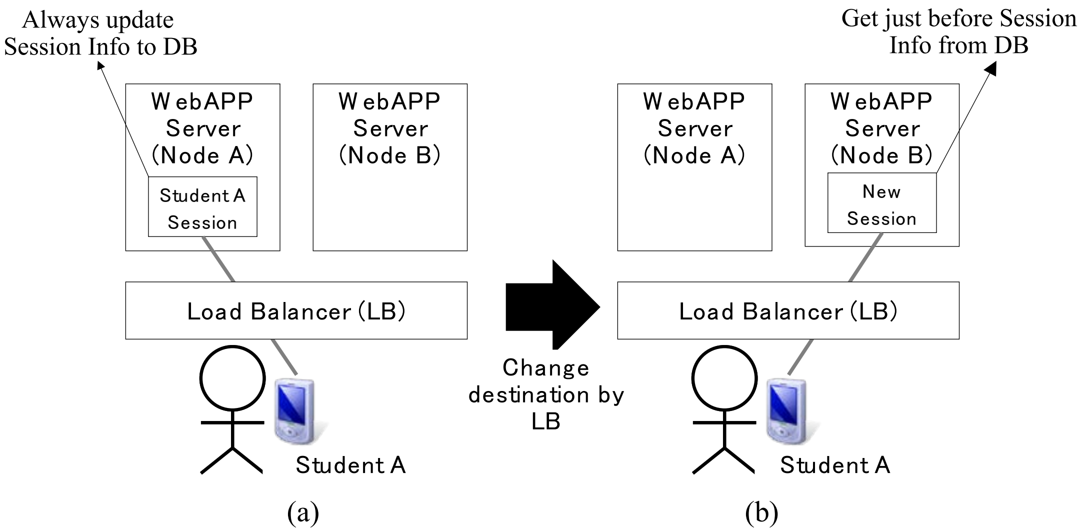

Figure 3. Mechanism of Session Info on WebCRS.

Session Info is always stored to the DB to enable current Session Info to be shared by each web server node even when the LB changes connections. As a result, verification of the scalability of WebCRS confirmed that the effect of load distribution is reduced when the system is used simultaneously by over 400 students (Mizutani, 2015). An investigation attributed this decline in performance to the load caused by recording Session Info to the DB. When using the Activity Log for the learning support function, the load of the DB is increased, thereby increasing the possibility of influencing the basic functions of the CRS. To realize the learning support function, a mechanism for using the Activity Log capable of maintaining system scalability is needed.

3.3 Appling the Concept of an Agent-Oriented System

This study attempted to resolve the above-mentioned problem by applying the concept of an Agent-Oriented System (AOS) to the system architecture of WebCRS. The AOS concept employs a Multi-Agent, which is an aggregation of an agent as an element of the system. An important concept of the AOS is a Mobile Agent, which is an agent capable of moving on networking, and is defined in FIPA as a standard specification of AOS (FIPA 98).

Figure 4 shows an example of system behavior adopting a Mobile Agent. The Multi-Agent layer is cross-sectionally employed in each node of the web servers. In each node, an agent that processes Session Info is created each time a student connection is established. The agent is a Mobile Agent that is able to move on the Multi-Agent layer; thus, if the destination of the connection is changed, the agent moves to a new node. Because it is able to move by retaining the Session Info, it becomes unnecessary to obtain Session Info from the DB. Furthermore, it also becomes unnecessary to always store Session Info to the DB, thereby reducing the load of the DB.

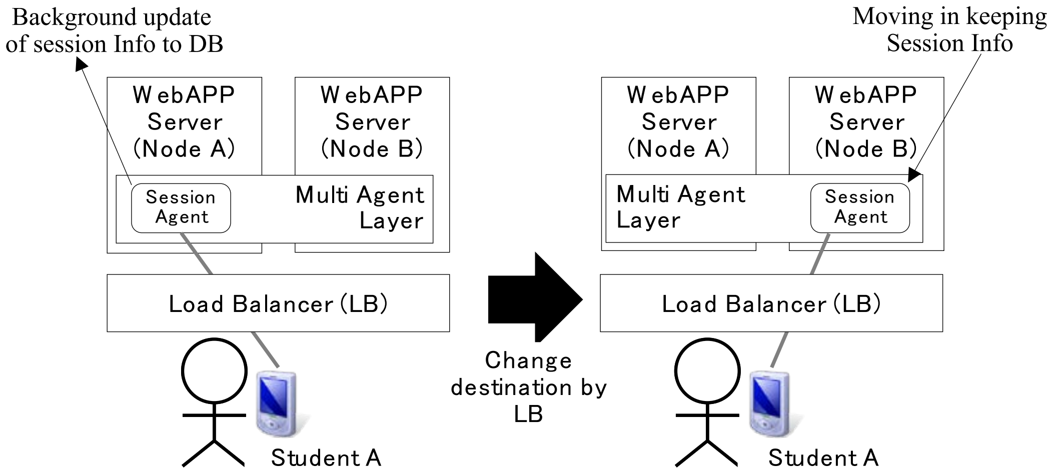

Figure 4. System structure of WebCRS applying the concept of AOS.

3.4 Prediction Using Activity Log for the Learning Support Function

To realize the automatic learning support function, three decisions need to be made: timing of support, a decision whether to provide support, and a method of support. For example, in relation to the timing of support, a function to provide supplementary teaching materials support includes a method to provide timing when a student answer is incorrect. However, this method may allow a student who answers randomly without thinking about a question to unintentionally receive supplementary teaching material. As to whether to decide to provide support, it is thought to provide the materials when the average of the correct answer of a student is less than a threshold that is configured beforehand. However, because the difficulty of each question varies, it is not possible to effectively decide by using a simple threshold only. Ideally, the educational effect would be enhanced by providing materials before a student submits their answer only when the possibility exists that their answer is incorrect.

In addition, it is necessary to realize this by reducing the load of the DB. Consequently, this study tries to predict whether the N th answer of a student is correct or incorrect by using the Activity Log of until the N-1 th answer, assuming to use the log maintained by the agents.

In this prediction, the following items in the Activity Log are used:

- Correct / Incorrect value of a response (*Cr*)
- Response Time required for each response (*Rt*)
- Label of limit time of each question (*Limit*)
- Activate Time for an response (*Act*)
- Flag value that means the time of a response is shorter or longer than the average of all students. (*ActF*)

Figure. 5 illustrates these items. *Rt* is the time between the moment a student's device received the message to start answering and the moment the student answered. Because of the time limit for answering each question, it is normalized into the range [0, 1]. *Limit* is a label that signifies the actual length of the time limit for answering as indicated in Table 1.

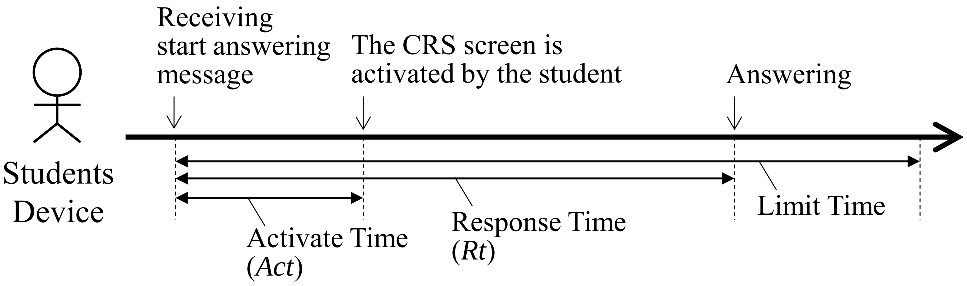

Figure 5. The items of Activity Log in the prediction.

Table 1: Labels indicating the Time Limit for answering.

Labels of *Limit*	Actual time limit for answering (sec.)
A	$t \leq 20$
B	$20 < t \leq 40$
C	$40 < t \leq 60$
D	$60 < t \leq 120$
E	$120 < t$

Act is the time between the moment the answer started and the time at which the CRS screen is activated by a student. Irresponsible students may sometimes be shown a screen other than the CRS. These students activate the screen of the CRS after they start answering and answer a question immediately after activation without thinking about the question. Because such student behavior might influence the correctness of an answer, *Act* is included in prediction data. Similar to *Rt*, *Act* uses the value that is normalized by the time limit for answering each question.

The purpose of the prediction based on the prediction result, depending on whether the *N* th answer of a student is correct or incorrect, it is to provide materials before the *N* th answer only when the possibility exists that the student answer is incorrect. At the moment just before the *N* th answering event of a student, it is not possible to know whether the *N* th answering time of the student is faster or slower than the average of the *N* th answering time of all students. In this case, *ActF* uses the values until the *N*-1 th only for prediction.

The Activity Log of *N* times of answering forms one set of data, $DataSet_i$, which is defined as follows.

$$DataSet_i = \{Cr_1, \dots, Cr_N, Rt_1, \dots, Rt_N, Limit_1, \dots, Limit_N, Act_1, \dots, Act_N, ActF_1, \dots, ActF_{N-1}\}$$

Here, *i*=1, 2, ..., *I*, *I* is the number of datasets that can possibly be used for prediction. For example, when the number of all answering events that are stored as the Activity Log is 6, if *N*=5, it becomes *I*=2. That is, the aim of this study is to predict Cr_N.

Furthermore, *Cr*, *Rt*, *Limit*, and *Act* are data that the agent is able to hold without access to the DB on the system model that applies the AOS. Because *ActF* is a flag to indicate the average answering time of all students, it is necessary to calculate the answering time that is held by each agent. This can be implemented by using the Blackboard Model of AOS.

4. Prediction Experiment

The above-mentioned items are predicted by using the Activity Log that consists of actual data in classrooms that introduced CRS. The Activity Log of this study consists of data obtained by using the "Response Analyst" in classes of students who attended an information technology course in our university. The course syllabus includes solving past versions of the Information Technology Passport Examination (IT Passport Exam), one of the Information Technology Engineers Examinations in Japan. The IT Passport Exam consists of multiple-choice questions. Examinees select one answer from among four choices in response to each question.

A total of 1,863 Activity Logs were stored. This included the logs of 53 students who attended the course, 50 questions from past versions of the IT Passport Exam, and the data concerned with this usage. Using these Activity Logs, the dataset for prediction is created. Three datasets are created by changing the value of N. The number of datasets I in each N is shown Table 2. These datasets are divided randomly according to a ratio of 3:1 for two purposes: (a) for training a classifier, (b) for testing the classifier.

The target of prediction is a correct or incorrect N th answer. This study uses two classifiers for the prediction: (1) A two-class neural network that has three fully connected hidden layers, (2) a two-class locally deep support vector machine (Deep SVN, Jose 2013). The neural network model of (1) is shown in Figure 6 and the result of the prediction is shown in Table 3.

Accuracy is the rate of correctly predicted results to the testing data. These are including the cases a student answer becomes correct and the cases it becomes incorrect. Precision is the rate of correctly predicted results to the predicted results that are a student answer becomes correct. Recall is the rate of correctly predicted to actual correct answers in the testing data. F-Score (F-measure) and AUC are indicators using these values, when these become to be 1.0, it represents the prediction result is good.

Table 2: Number of datasets.

N	I
3	676
4	523
5	400

```
input Data auto;
hidden Layer1[100,2] from Data all;
hidden Layer2[50,1] from Layer1 all;
hidden Layer3[25,1] from Layer2 all;
output Result[2] from Layer3 all;
```

Figure 6. Definition of neural network for prediction, written in Net#.

Table 3: Results of the prediction.

Classifier	N	Accuracy	Precision	Recall	F-Score	AUC
Deep SVN	3	0.568	0.560	0.659	0.605	0.568
	4	0.550	0.566	0.671	0.614	0.552
	5	0.580	0.608	0.585	0.596	0.561
Neural Network (NN)	3	0.497	0.000	0.000	0.000	0.560
	4	0.649	0.633	0.814	0.713	0.640
	5	0.510	0.531	0.642	0.581	0.520

5. Consideration

To realize an automatic learning support function, this study considered applying the concept of AOS to WebCRS. It is thought that the effect of load balancing is improved and the load of the DB system is reduced. As a result, it is possible to realize automatic learning support functions using the Activity Log of students stored in the DB. On other hand, the compatibility between an agent framework and the web application framework used by WebCRS presents a problem. In the current state, WebCRS is able to operate on SaaS type public cloud services. Major SaaS frameworks do not support the implementation of agent frameworks such as JADE (Bellifemine, 1999). Although this problem can be resolved by the implementation of unique mechanisms, the compatibility might decrease. Therefore, it is necessary to consider a method for the implementation of an agent framework to maintain compatibility with cloud services.

In terms of the experimental results for the correct / incorrect prediction of answers using an Activity Log, the best result was the case of N=4 when using a neural network (NN). It shows that it is able to predict the answer of a student, with a probability of 0.649. When N=3, the NN resulted in overfitting. When N=5, the result decreased in comparison to the case of N=4. The reason for this is thought to be that the number of datasets is smaller than the case of N=4. In addition, this study did not classify the dataset according to each student. Because the time required for providing a correct answer and the learning attitude are different for each student, the prediction results are expected to improve when the dataset is classified according to each student. However, this would increase the size of the Activity Log of each student.

The probability of an answer being correct or incorrect is 0.5. The Activity Log used in this experiment involves responses that consist of one answer from among four choices in a question; thus, the probability of a correct answer is 0.25. And also, the actual ratio of correct answers of the all students is 0.523. Each item of the Activity Log using the prediction does not have a direct logical relation to each other. However, when using CRS, it is possible to indirectly interpret the extent to which a student understands the learning material as the time they require to answer, their behavior in terms of switching screens to and from other application, and so on. Because the probability of the results including DeepSVN is over 0.5, it is possible to predict whether an answer is correct beforehand. By improving the accuracy of the prediction, it is thought that a function providing supplementary teaching materials before the student answers when an incorrect answer is predicted can be realized.

6. Conclusion

This paper discussed the problems of conventional CRS, described plans for resolving the problems and to implement an automatic learning support function. On web-based CRS, a learning support function that adapts to the state of student answering could be realized by applying the concept of AOS as system architecture. Effective learning support was achieved by attempting to predict a student answer by applying the architecture.

By applying AOS, it is possible to realize automatic learning functions while maintaining the scalability and usability as a basic function of CRS. Although the accuracy of the prediction results was insufficient, the possibility of realizing an automatic learning support function that predicts whether a student answer is correct or incorrect before they answer, and automatically provides supplementary teaching materials was described.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP15K16262.

References

Barth-Cohen, Lauren A., Smith, Michelle K., Capps, Daniel K., Lewin, Justin D., Shemwell, Jonathan T. & Stetzer, MacKenzie R. (2016) , What are Middle School Students Talking About During Clicker Questions? Characterizing Small-Group Conversations Mediated by Classroom Response Systems, *Journal of Science Education and Technology*, Vol.25, No.1, pp.50-60.

Bellifemine, F., Poggi, A. and Rimassa, G. (1999), JADE – A FIPA-compliant agent framework, *Proceedings of the Practical Applications of Intelligent Agents*, pp.97-108.

FIPA (1998), FIPA 98 Specification Agent Management Support for Mobility, Foundation for Intelligent Physical Agents.

Fujita, H., Hirata, K., and Sato, T. (1969), Theoretical Investigation on the Response Analyzer for Group Education, *Trans. Inst. Electron. Commun. Eng. Japan*, Vol. 52-C, pp.820-826.

Hunsu, N. J., Adesope, O., and Bayly, D. J. (2016), A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect, *Computers & Education*, Vol. 94, pp.102-119.

Jose, C., Goyal, P., Aggrwal, P., Varma, M. (2013), Local Deep Kernel Learning for Efficient Non-linear SVM Prediction, *JMLR W&CP,* Vol.28, No.3, pp.486–494.

Karakostas, A., Adam, D., Kioutsiouki, D. and Demetriadis, S. (2014), A pilot study of QuizIt: The new android classroom response system, *2014 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2014)*, pp. 147-151.

Lantz, E. M., Stawiski, A. (2014), Effectiveness of clickers: Effect of feedback and the timing of questions on learning, *Computers in Human Behavio*r, Vol.31, pp.280-286.

Liu, C., Chen, S., Chi, C., Chien, K., Liu, Y. and Chou, T. (2016), The Effects of Clickers With Different Teaching Strategies, *Journal of Educational Computing Research*.

Majumdar, R. and Iyer, S. (2015), Beyond Clickers: Tracing Patterns in Students' Response through iSAT, *Proceedings of the 23rd International Conference on Computers in Education*, pp.612-621.

Mizutani, K. (2013). Development of Response Analyzer Systems improving scalability and real-time processing. *Proceedings of the 38th Annual Conference of Japanese Society for Information and Systems in Education*. TE2-3, 323-324.

Mizutani, K. (2016), A Study for Using Sensors of a Smart Device on Web Based Classroom Response Systems, *Proceedings of the 15th Forum on Information Thechnoogy Japan*, N-021.

Mizutani, K.(2014), A Study of Student Behavior in Classroom Response Systems, *Proceedings of the 22nd International Conference on Computers in Education*, pp.483-488.

Premkumar, K. (2016), Use of Student Response Systems for Summative Assessments, *Creative Education*, Vol.7, pp.1851-1860.

Richardson, A. M., Dunn, P. K., McDonald, C. and Oprescu, F. (2015), CRiSP: An Instrument for Assessing Student Perceptions of Classroom Response Systems, *Journal of Science Education and Technology*, Vol. 24 (4), pp 432-447.