

A System for Strategic Exercise of Cooperative Software Design Focusing on Relationships between the Components of Technical Documents

Shoichi Nakamura, Gotoh Fumihiko, Hiroaki Kaminaga, and Youzou Miyadera

Abstract— This project aims to achieve novel software design exercises for cooperative technical document editing in which relationships between technical documents are considered. This paper mainly describes the design of a support system together with an outline of a strategic exercise. It also explains the algorithm for diagnosing the relationships between the components of technical documents. Moreover, this paper also describes the development of main functions. This system is equipped with support functions for reviewing the relationships between the components of documents and a visual environment for sharing editing contexts.

Keywords— Cooperative software design exercise, Relationships between documents, Editing context, Context sharing, Technical documenting

I. INTRODUCTION

DEMAND for engineers who can design and develop software has increased in recent years. They mainly obtain their skills and knowledge on cooperative software design through experience. Many universities therefore offer exercises on cooperative software design and development for training potential engineers with such knowledge. However, it is actually difficult to learn sufficiently in these classes because of constraints peculiar to them.

There exist support systems for software design exercises (e.g. [10]) and systems for assisting cooperative reviews (e.g. [5]). However, most existing systems do not sufficiently consider the relationships between the technical documents from the aspects of training beginners.

This research has developed a support system for cooperative software design exercises in which relationships between the documents are considered and editing contexts are shared together with strategic learning flow. This system achieves practically effective cooperative software design exercises.

Shoichi Nakamura, Gotoh Fumihiko, and Hiroaki Kaminaga are with Department of Computer Science and Mathematics, Fukushima University, Japan.

Youzou Miyadera, is with Division of Natural Science, Tokyo Gakugei University, Japan.

II. TARGET AND APPROACH

A. Target Exercise and Problems

This research targets exercises that aim to give learners empirical knowledge such as how to create documents by using shared editing contexts, interaction among members, and paying attention to the relationships between the documents.

We initially investigated the actual cooperative software design exercises in universities and clarified the following problems to overcome.

- (1) It is difficult for beginners to learn about cooperative design processes while paying attention to the relationships between the documents.
- (2) It is difficult to share the editing contexts such as intentions during document polishing.

B. Approach

To solve these problems, this research initially designed a strategic learning flow to make learners experience technical document creation while paying attention to the relationships between the documents, sharing editing context, and polishing documents on the basis of it. For this, the following two diagnosis methods were developed.

- A method for diagnosing the relationships between the technical documents in component level and in phrase level.
- A method for enhancing sharing of the editing contexts among learners.

By introducing these methods, a support system for a strategic exercise of cooperative software design was developed. This system provides a smart environment for context sharing by easy annotation and document polishing based on it. Moreover, this system alerts learners to suspicious description parts with wrong relations between the documents and their components.

III. RELATED RESEARCH

Many research projects have aimed to support software design [1]-[6]. However, these projects were not effective for exercises since they did not consider the training of beginners. Management systems exist for reviewing designs using the

meeting minutes [5] and a Web-based environment for review [6]. However, these systems cannot be used for beginner exercises since the composition of technical documents and the considerations of review are separately managed. Moreover, these existing systems are not applicable for supporting exercises because these projects have not considered the restrictions and characteristics peculiar to exercises.

Support systems exist for software design exercises [7]-[11]. Some of these existing systems provide practical functions based on the consideration of restrictions in exercises. However, these systems have not provided direct support for training in the composition or polishing of technical documents. Although some systems focused on communication among members of a cooperative design team, these systems are difficult to apply for synthetic support of technical document creation and sharing of editing contexts since most systems adopt BBS as the environment of communication.

On the other hand, there are systems to support the sharing of know-how among learners [12]-[15]. These systems are effective for knowledge sharing and discussion support among learners. However, they are not effective enough for cooperative technical composition in software exercises because they separately manage the information sharing and discussion.

This background has motivated us to develop a support method for cooperative software exercises so that beginners can learn technical composition and document polishing while considering the relationships between the documents and sharing editing contexts.

IV. SUPPORT SYSTEM FOR SOFTWARE DESIGN EXERCISE FOCUSING ON COOPERATIVE TECHNICAL COMPOSITION

A. Overview

This system is designed as a component of a synthetic environment that supports the design process management, products of design (e.g. technical documents, diagrams, program sources), and editing contexts. Learners conduct the exercises in groups with the project theme predefined by a professor. They work mainly on composition of technical documents, discussion, and polishing of the documents on the system. Technical documents are submitted and related information such as editing contexts is managed also in the support environment. This system diagnoses the correctness of the relationships between the submitted technical documents and their components. If any problems and suspicious points are found, the system indicates them to the learners. In this research, to start with, the technical documents listed in Table I are assumed as targets.

Generally, there are important relationships between the technical documents handled in the projects. Moreover, there are also meaningful relationships between the components of the document. Therefore, it is important for learners to grasp these relationships precisely. However, beginners are often unaware of such relationships since they tend to concentrate only on filling in the document. On the other hand, simply

providing beginners with the completely extracted relationships between the documents is not the best way to train them. In this research, we make learners associate the documents and their components by themselves as one of the steps of the exercise. This task focuses learners' attention on the relationships and their importance. In addition, this system indicates wrong associations so that beginners can learn technical composition methods together with correct relationships.

TABLE I
TARGET TECHNICAL DOCUMENTS

Design process	Technical document	Outline of document
Basic design	Requirement specifications	Workflow, concept model
<i>External design</i>	Overview design	Requirements Overview, function overview
<i>External design</i>	Screen design	Screen transition diagram
<i>Internal design</i>	Function design	Function list, details of function
<i>Internal design</i>	Database design	ER-diagram, definition of tables

B. System Requirements

On the basis of observations of actual software design exercises, requirements of this system are arranged as follows.

- This system should support composition, editing, and management of technical documents.
- This system should provide the environment for annotations to the technical documents and their management.
- This system should help users to grasp the edited descriptions in the documents and their footprints.
- This system should help users to grasp the relationships between the technical documents and their components.

C. Function Design

To satisfy the requirements, the system is equipped with the following functions along with the basic functions.

- Function for composition and editing of the document.
- Function for sharing the editing contexts
- Function for associating the documents and their components
- Function for diagnosing the relationships between the documents and their components.

D. Diagnoses of the relationships between technical documents

When learners relate the technical documents after their creation, the system diagnoses those connections between the documents in their item level. The system compares the relating by a professor and by learners. Specifically, the system examines the following points.

- (1) Whether learners perfectly marked the important items defined by a professor.
- (2) Correctness of each connection (departure and destination) between document A and B.

If there are any wrong or missed points, the system alerts learners of them. Learners check and revise their relating on the

basis of the alerts until everything is correct. This activity enhances learners' understanding of the relationships between the documents in a basic level.

After that, learners relate the technical documents by marking important phrases and connecting them. After this relating, the system carefully diagnoses the description part with the

following three situations.

- (Situation.1) Whether learners perfectly marked the important phrases defined by a professor. (Fig.1)
- (Situation.2) Mismatch of connection in the item level (Fig.2).
- (Situation.3) Wrong destination of each connection in document B (Fig.3)

(Situation 1) Overlooking of important phrases

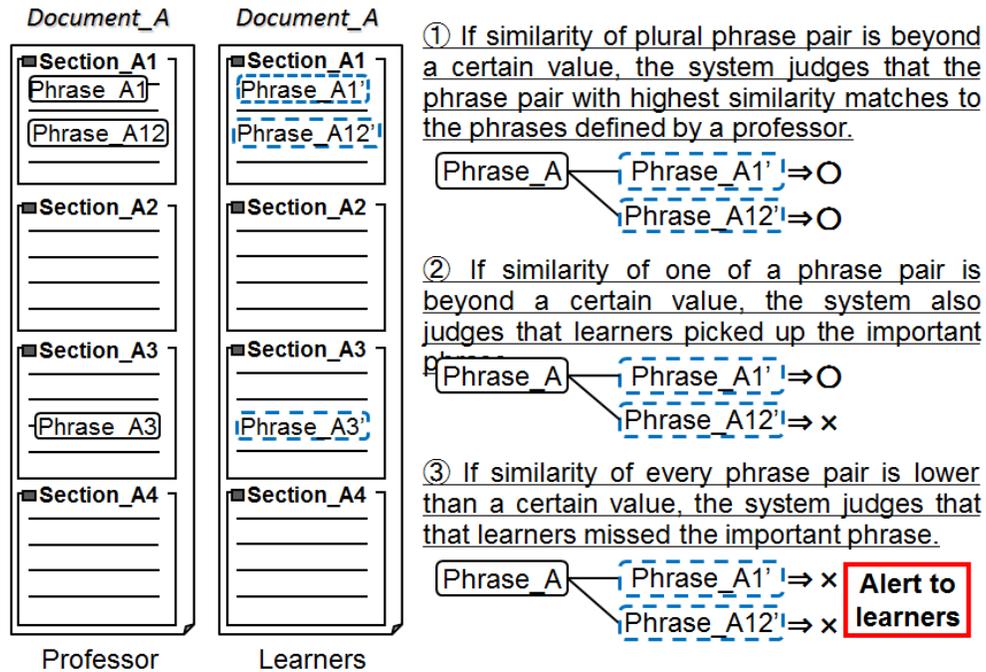


Fig. 1 Diagnosis of overlooking of important phrases

(Situation 2) Mismatch of connection in item level

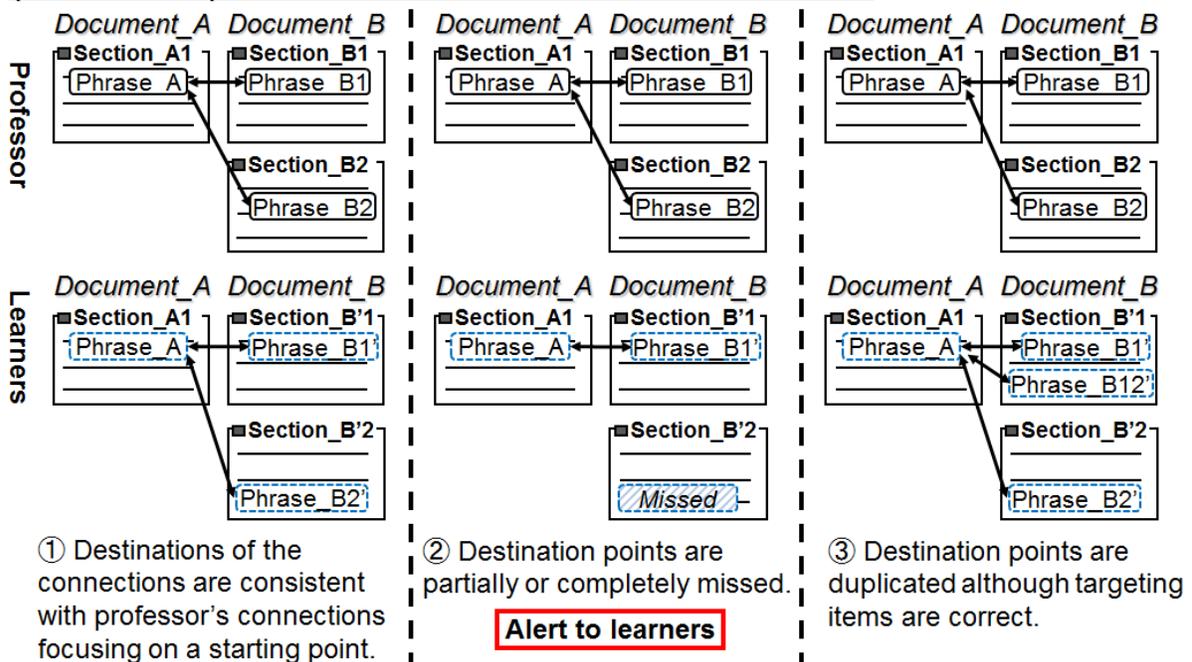


Fig. 2 Diagnosis of mismatch of connection in item level

(Situation 3) Wrong destination of each connection in document B

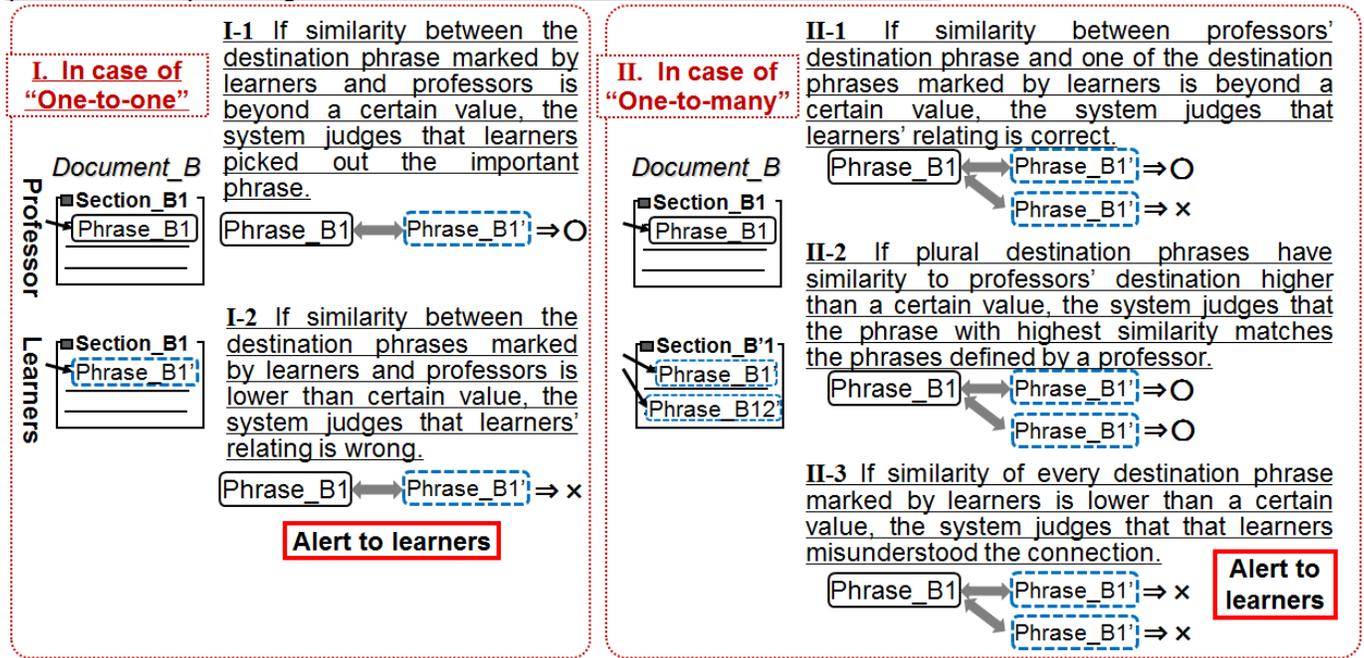


Fig. 3 Diagnosis of wrong destination of each connection in document B

If there are any suspicious points, the system visually alerts them to learners with color coding. Learners check each alert one-by-one and revise it with trials and errors. This activity enhances learners' understanding of the relationships in detail.

E. Develo~~p~~ment

Function for composition of the document and their editing

This research provides a system environment for editing and managing technical documents on the Web. A professor uses this function when he/she composes the sample of technical documents for preparing the exercise. Learners also utilize this function when they compose the documents in the process of the exercise.

This function enables easy composition of the documents by inputting the contents using the visual forms after selecting the target kind of documents from the predefined lists. Learners are able to add the sections and more detailed items by easy operation with UI (Fig.4). Learners are able to confirm the latest contents at a preview provided by the system (Fig.5). This function thus achieves an intuitive and comfortable environment for editing.

Function for sharing the editing contexts

Function for marking: It is pointed out that visual marking of a description in the document is effective for satisfactorily polishing the document and sharing information, especially for cooperative works in a distributed environment [12]. This research therefore provides smart tools for marking to encourage learners to mark noteworthy parts in the documents (Fig. 6). Marking provides a visual impact that contributes to

awareness.

Editing operations and updates of technical documents necessarily involve investigating the contexts, e.g. author's intention, opinion, and noteworthy descriptions. Such contexts are quite important for building a common understanding among members and for keeping technical documents consistent. Therefore, the contexts must be satisfactorily accumulated in an easy-to-utilize style.

This function enables the annotation of the technical documents to preserve the author's intensions and questions together with the circumstances. This function also enables the sharing of the annotations among members. Annotated descriptions are highlighted automatically, and learners can check them just by clicking (Fig. 6). This view enables the circumstances of editing to be easily understood.

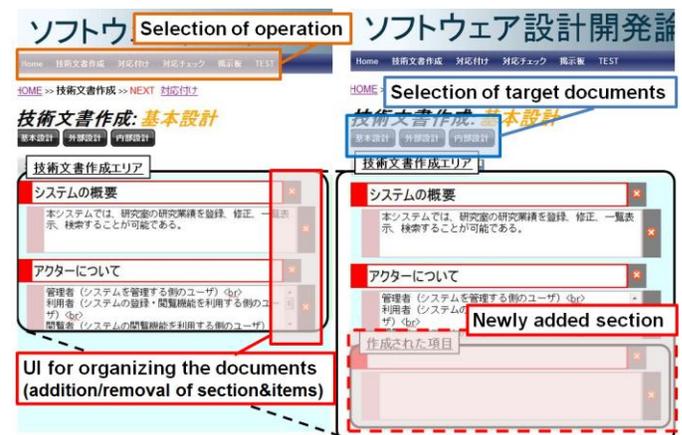


Fig. 4 Organizing the documents by adding sections using visual UI

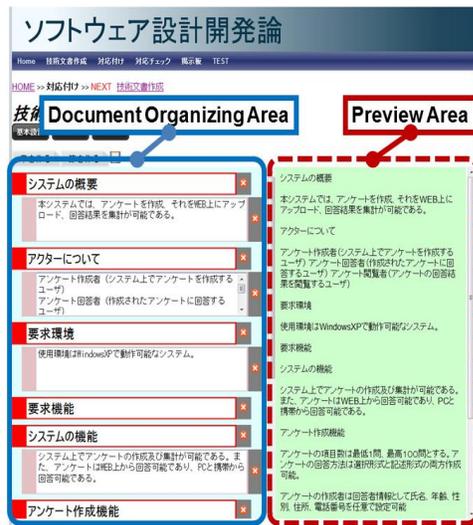


Fig. 5 Visual document organization and preview

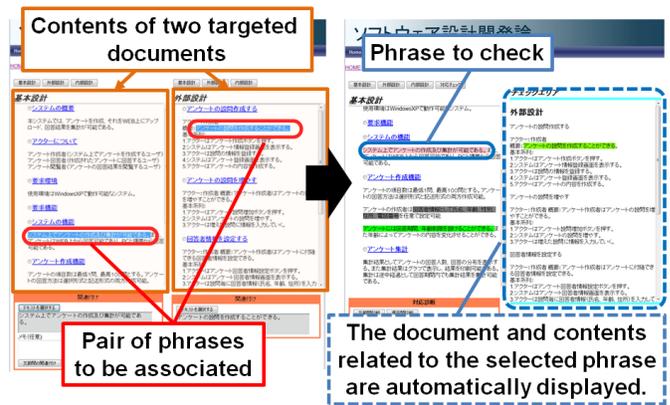


Fig. 7 Associating the documents and their components

Through these associations, learners are also able to annotate the circumstances of association using the function for sharing the editing contexts. Moreover, this function enables a professor and learners to confirm the latest associations and their targets intuitively.

Function for diagnosing the relationships between the documents and their components.

This system diagnoses the relationships between the technical documents and their components on the basis of the comparison between samples of the professor and products of learners. If wrong associations are found, this function alerts learners of the suspicious points. Receiving the alert, learners reconsider the relationships between the documents together with their contents and make revisions.

F. Flow of Strategic Exercise

Fig. 8 outlines the flow of an exercise using our system. Initially, a professor prepares a pair of sample documents: document A is created faster and is referred to in the other document, and document B is created later and refers to document A.

As a first step of exercise, learners carefully read document A and mark the important phrases. Then, a support system diagnoses the coincidence of phrases marked by learners and by a professor. If overlooking of important phrases is found, the system alerts it to learners. Learners reconsider the important descriptions in document A with reference to suggestions by the system.

After understanding the important descriptions in document A, learners connect the appropriate items that build a pair of documents prepared by a professor to grasp the relationship between the items. A support system indicates the differences in relating by learners and by a professor on the basis of their comparison. A learner repeatedly revises the relating until relationships between all items meet those related by a professor. Learners can understand the relationships between the document A and B in the item level until he/she completes the revision.

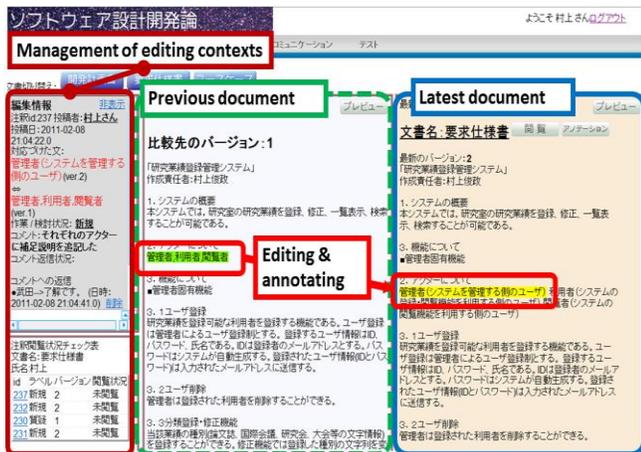


Fig. 6 Annotating and editing context sharing

Function for associating the documents & their components

This function achieves an environment in which learners associate the related technical documents and their components. After composing the documents, learners select the target documents of associating. Then, our system visually presents the two target documents side-by-side (Fig. 7). This system supports association in both the item level and the phrase level.

In the case of association in the item level, learners initially select the target item using visual UI. Then, learners decide the destination of the association after considering the organization of the documents. After these operations, the selected items are associated and their information is registered into databases.

In the case of association in the phrase level, learners initially specify the scope of contents by mouse dragging. Learners put this selection into the two targeted documents and fix the association by simple UI operation. Then, the two specified phrases are associated, and their information is automatically registered into databases.

Then, a learner creates document B using only the items prepared by a professor with reference to document A. After the document creation, a learner relates the document A and B by connecting their phrases. In this step, a learner annotates his/her intention and points to be investigated into the marked descriptions for relating. Then, the system manages these annotations in a database and visually presents them to learners. This visual presentation alerts members to annotations inserted by a learner and encourage members to share the editing contexts and polish the document on the basis of them. This support becomes more effective for practical exercises, which include the division of roles and asynchronous work.

The support system also diagnoses the differences between the document relationships in phrase level drawn by a learner and by a professor. If suspicious parts are found, the system alerts them to learners. Learners can understand the relationships between the components of document A and B in phrase level by repeatedly conducting this work.

These three strategic works (marking of important description, documents relating in the item level, and documents relating in phrase level) while sharing the editing contexts enable learning through practical experience.

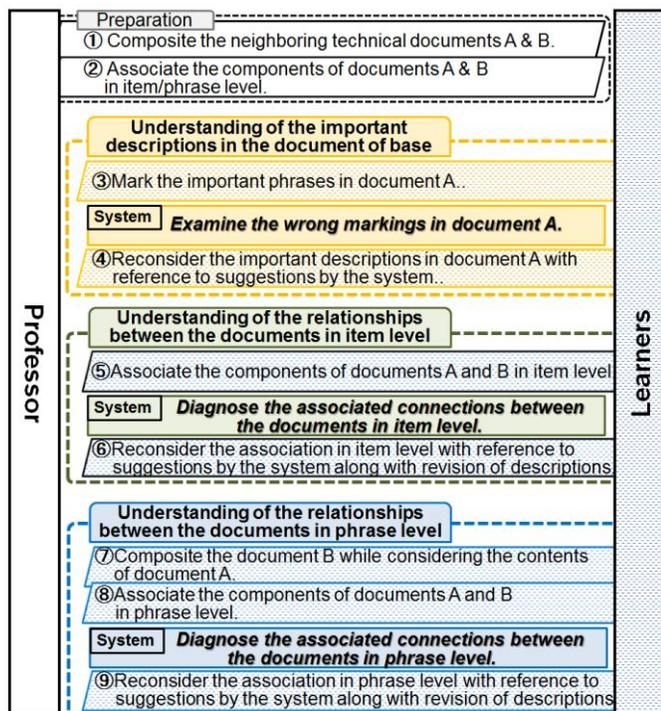


Fig. 8 Flow of strategic exercise of cooperative technical documenting

V. CONCLUSION

This paper described the design of a support system for cooperative technical composition together with strategic learning flow. Developments of main functions were also described. Our system provides a smart environment for easy editing of technical documents and context sharing in its processes. This system also diagnoses the relationships between the documents and alerts learners to suspicious descriptions with wrong associations.

Future works include careful examination of the methods and diagnosis criteria together with the strategy of an exercise.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Education, Culture, Sports, Science and Technology, Japan and by Japan Society for the Promotion of Science under Grant-in-Aid for Challenging Exploratory Research (No. 24650544), Young Scientists (A) (No.25702007) and Scientific Research (B) (No. 25282051) .

REFERENCES

- [1] B. A. Farshchian, "Gossip: An Awareness Engine for Increasing Product Awareness in Distributed Development Projects," in *Proc. the 12th International Conference on Advanced Information Systems Engineering (CAiSE 2000)*, pp 264-278, 2000.
- [2] K. R. Al-asmari, and L. Yu, "Experiences in Distributed Software Development with Wiki," in *Proc. 2006 International Conference on Software Engineering Research and Practice*, pp.389-393, 2006.
- [3] R.J. Ocker, "Influences on creativity in asynchronous virtual teams: a qualitative analysis of experimental teams," *IEEE Trans. Professional Communication*, Vol.48, pp. 22-39, 2005. <http://dx.doi.org/10.1109/TPC.2004.843294>
- [4] R.J. Ocker, S. Roxanne, H.M. Turoff, and J. Fjermestad, "The Effects of Distributed Group Support and Process Structuring on Software Requirements Development Teams: Results on Creativity and Quality," *Journal of Management Information Systems*, Vol. 12, No. 3, pp. 127-153, 1995.
- [5] Y. Kudo, C. Hirai, H. Kawabe, Y. Huruhashi, and O. Ohno, "Development and Evaluation of a Minutes-based Review Management System for Software Project," *IPSJ Journal*, Vol.44, No.5, pp.1404-1412, 2003.
- [6] S. Komiya, "The Introduction of Kepner-Tregoe Program into Software Trouble-Shooting Performed on WWW and Improvements in It," *Trans. IEICE*, Vol.J81-D-2, No.5, pp.992-1004, 1998.
- [7] A. Hazeyama and A. Nakano, "Communication Support System for Software Engineering Education," *IPSJ Journal*, Vol.42, No.11, pp.2550-2561, 2001.
- [8] L. Zhang and A. Hazeyama, "A Collaborative Learning Support System for Software Engineering Education," in *proc. the 6th Annual Conference on Collaboration Technologies*, pp. 123-126, 2012.
- [9] R. Kawai and A. Hazeyama, "A Know-How Recommendation System for a Software Engineering Project Course By Using the Content Filtering Technique," in *Proc. the 34th Annual International Computer Software and Applications Conference (COMPSAC2010)*, pp. 547-548, 2010.
- [10] A. Hazeyama, "A Case Study of Undergraduate Group-based Software Engineering Project Course for Real World Application," in *Proc. the First International Symposium on Tangible Software Engineering Education, (STANS2009)*, pp. 39-44, 2009.
- [11] K. Nishida, M. Kotani, and K. Ochimizu, "Support System for Information Sharing in Distributed Cooperative Software Development," *Information Processing Society of Japan, SIG-SE Technical Reports*, Vol.2004, No.30, pp. 17-23, 2004.
- [12] T. Hayama, X. Yang, and S. Kunifuji, "A Collaborative Notebook System that Provides Learning-knowledge Filtering," *IPSJ Journal*, Vol.48, No.8, pp.2814-2822, 2007.
- [13] D. K. O'Neill and L. M. Gomez, "The Collaboratory Notebook: a Networked Knowledge-Building Environment for Project Learning," in *Proc. ED-MEDIA94*, pp.416-423, 1994.
- [14] D. C. EDELSON and D. K. O'Neill, "The Covis Collaboratory Notebook: Supporting Collaborative Scientific Inquiry," in *Proc. the NECC94*, pp.146-152, 1994.
- [15] K. Umeda, T. Yasuda, and S. Yokoi, "Proposal of a Research Information Sharing System Utilizing Knowledge-Memos," *IPSJ Journal*, Vol.42, No.11, pp.2562-2571, 2001.