We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Realization of Swift and Expressive Chat in CSCL: Reach Project

Wataru Tsukahara¹, Shingo Akamatsu², Xin Wan² and Toshio Okamoto²
Tokyo University of Agriculture and Technology¹, University of Electro-Communications²
Japan

1. Introduction

A Computer Supported Collaborative Learning (CSCL) is a learning where learners make group and learn through discussion with the help of computers (Koschman 1996, ISO/IEC JTC1 SC36 WG2 2004). In CSCL, learners interchange their knowledge as discussion goes on. In addition, CSCL is expected to help learners acquire good communication skills for smooth discussion such as modifying explanation according to the other members' knowledge.

Many Learning Management Systems (LMSs) are equipped with text chat tool and electric bulletin board systems (BBS), or Mailing List (ML). They are slightly different from each other (Tamura, 2005). In BBS/ML, the period of discussion is long and content tends to be logical. In text chat, session is short and casual topic is favored. Abrams (2003) has compared chat and BBS in language learning, and reported that CSCL discussion must have both the features. Short sentences are frequently exchanged but they must be logical. Otherwise it will not be learning. Thus, it is better for CSCL discussion tools to help learners smoothly continue logical discussion.

Fig. 1. Illustration of problems in chat

One problem about smooth discussion is congestion. In face-to-face speech dialog, backchannels, fillers, and overlapping speeches effectively work for smooth turn taking (Duncan & Fiske 1977, Sacks, Schegloff & Jefferson 1974). However, in text chat, each

sentence is sent to others only after completion. This lack of real-time communication easily causes congestion. Moreover, text chat tools cannot use non-verbal cues such as disfluency, pitch, loudness, and speech rate (Kitade 2000), which can transmit confidence, doubt, and interest in parallel with discussion itself (Ward 2004). While some spoken Dialog Systems can use non-verbal cues to improve computer human interaction (Tsukahara & Ward 2001), text chat has difficulty of knowing other members' affect by the lack of non-verbal cues. Some examples of problematic situation are illustrated in Figure 1.

There are some systems to tackle these drawbacks. The classic 'talk' program in UNIX can display letter by letter. It is simple and easy to use but usually used for casual chat. Yamada & Takeuchi (2005) proposed a system that displayed input letter in real time for multiple users. In the system, utterances are scrolling and fading away like in speech situation. This inconvenient implementation is because they focus on social interaction, rather than learning. Hayashi et al. (2004) developed a new chat system 'Synchat' and solved this problem by displaying real time key typing in usual chat window. They attached the display of currently typed in characters by others below the usual chat log displaying text area. They reported that using the system reduced simultaneous key typing by multiple users and users felt they became careful about other users' opinion.

The second problem, about logical discussion, seems to be a casual/formal or a private/public switching problem. Cook et al. (2005) developed a CSCW collaborative java code editor and collaborative UML diagrammer for pair coding. The cursor of the buddy is highlighted and position in source code is updated in real time. The system improved overall effectiveness and users felt they better understood others' changes. However, they commented the need of "private work area" for editing. This implies users want to work around before updating. This aspect is not so much regarded as important in Synchat (Hayashi et al. 2004) because input by all users are displayed in a compounded form in one area below usual chat log display area. This may good for keeping focus of discussion, but freedom of editing is implicitly restricted by the area of physical geometry. Sometimes learners want to edit more than one line, and it would be better they can enlarge their editing space freely. Kienle (2008) developed a system KOLUMBUS2 where BBS and chat are both available in a discussion. KOLUMBUS2 provides "clipboard" area for such a purpose. It was used in chat sessions with moderator where learners tended to post more elaborated ideas. Thus we think some kind of editing space is necessary for constructing sequence of logical ideas. Kienle also reported that through the evaluation of KOLUMBUS2, chat was less used in formal phase of discussion. We think one possible reason for this is because BBS and chat is so different in their appearance and learners thought chat was just prepared for casual, temporal exchange of ideas. One solution may be to make GUI of BBS and chat similar.

The third problem, about logical discussion is "focus". SubEthaEdit (Pittenauer & Wagner 2005) is perhaps the most successful collaborative editing environment in real use. It allows users simultaneous editing. Addition, insertion, deletion, and selection are highlighted in real time. Although SubEthaEdit is very effective, some features can be too rich. For example, each user can edit their own interested part and sometimes go out of window, which may cause division of discussion. A kind of "reminder of current focus" seems to be necessary. In collaborative discussion, participants (learners) must be able to edit their opinion freely. At the same time, it is better for all the participants to focus in the same topic. However, manipulating participants' focus directly may be difficult. Instead, we plan to provide

participants with “bird view” of discussion to help them be aware of focus of dialogs, at the same time providing free editing area with awareness.

From the above consideration, we have developed a text chat tool “Reach” (Tsukahara et al., 2007a). This chat tool shows real time typing process to all the participants. We have also developed some small gadgets for discussion analysis purpose. Furthermore, motivated from preliminary analysis, we have added a function of displaying face mark which corresponds to Dialog Act status.

This paper reports the detail of the system and results of preliminary evaluation, and then detail of modification by adding Dialog Act display function. In the next section the concept, features, and implementation of Reach are described. The third section describes a result of preliminary evaluation. The fourth section describes an additional function of Dialog Act display using face mark. The fifth section discusses the prospect of this work. We summarize our work in the last section.

2. Real Time Display of Key Typing Process

2.1 The REACH chat tool

A screenshot of Reach chat tool is shown in Figure 2. It uses two kinds of windows. The features of this tool are:

- Real time letter by letter transmission: for improving awareness of editing process (Chat window).
- Private workspace: participants/learners can feel free to write and delete for building ideas (Chat window).
- Past sentences reference: a window for displaying past sentences as discussion history. Participants/learners can check mainstream and consistency. This may also useful as centripetal force for current focus of discussion (Talk log window).

Fig. 2. Display of Reach (2 desktop is concatenated for convenience)

Table 1 shows comparison between proposed system (Reach) and other systems. The major difference from Yamada’s system (Yamada et al. 2005) is that the use of past sentences in “Talk log” window. When a participants/learner decides to submit a sentence in private workspace as an opinion, “Shift” plus “Enter” are pressed, and then it is displayed in the Talk log window. In this way, we explicitly divide two types of information in two separate windows. The difference from Collaborative tools (Pittenauer & Wagner 2005) and Synchat

(Hayashi et al. 2004) is the private workspace. In the private workspace, users can edit ideas freely because their sentences are still not present in log window before publicly registered. This window also works as an awareness display for turn taking and non-verbal cues such as hesitation, confusion, rush, boredom, etc.

	speech dialog	text chat	Yamada <i>et al.</i>	KOLUM-BUS2	SubEth>Edit	Synchat	Reach
purpose	casual chat, discussion	casual chat	casual chat	discussion	collaborative coding	discussion	discussion
modality	speech	text	text	text	multi color text	text	text
transmission	real time	after completion	real time	after completion	real time	real time	real time
private workspace	no	yes	yes	yes	no	one line	yes
past sentences reference	impossible	possible	impossible	possible	possible (not real time)	possible	possible
centripetal force for current discussion	strong	strong	strong	strong	weak	moderate	moderate

Table 1. Comparison between Reach and other text chat systems

Reach is a server and client system. It uses socket communication with original protocol on specified port. Major commands in the protocol are ‘CHAT’ and ‘TALK’. Each time a learner enters a key, the client transmits the current tentative sentence to the server by CHAT command, and then the server broadcasts it to other clients. These texts are displayed in CHAT window. When the learner completed the sentence by pressing “Shift” plus “Return” keys, it is sent to the server by TALK command, and the server registers the sentence to discussion log as in the same format as usual text chat systems. This is displayed in the Talk log window (Figure 3).

Fig. 3. Transmission of CHAT command and TALK command

Figure 4 shows a transmission log. The system transmits whole text strings, not only sending differences. We knew this is a very awkward way, however it makes

implementation much easier when we consider handling erasing region (Ctrl+X in Windows) etc.

Location of each user’s workspace is also transmitted in this protocol. This makes utterances such as “the top left guy seems confused” meaningful.

Fig. 4. Transmission of CHAT command and TALK command

2.2 Gadgets for post analysis

We also prepared small tools for post analysis of discussion. Details of these tools are described in (Tsukahara et al., 2007b).

Talk log viewer, developed by the second author, is convenient to replay the past discussion visually (Figure 5). This tool converts a talk log into an html file. In a browser window, discussion is displayed in longitudinal direction. Green region is typing period and blue region is silent period. If a user points the mouse on green region, typed letters are shown. User can trace the typing process by moving the mouse over the green region.

Fig. 5. Talk log viewer

Dialog Act Viewer, developed by the third author, is used to see to whom the utterance is directed (Figure 6). Dialog act direction is determined by specific mark such as “>> Tom” at the last of a sentence. For example, from a TALK sentence “I agree >> idoji”, this “agreement” sentence is toward the user ‘idoji’.

Typing speed calculator is a tiny java code for calculating the typing speed. This corresponds to speech mora rate in spoken dialog.

Fig. 6. Dialog Act viewer

3. User Evaluation

We conducted eight chat sessions to compare the difference between Reach and conventional text chat with seven learners (Subject A, B, ..., G). Table 2 shows the details. The first session was a pilot session, so that we did not set control condition using “Normal chat”

session	Topic	Reach user	Normal chat user	Length (h:mm)
1	Is sea-level rise caused by global warming (anathema)?	ABC	-	1:42
2	modify the form of chain rings by three action	DEF	BAC	0:24
3	How to find a black hat scholar in n scholars	DEF	BAC	0:38
4	each letter represents a digit: SEND + MORE = MONEY	DGA	EFC	0:25
5	How many days for snail to climb the well?	DGA	EFC	0:04
6	Make 100 from given numbers and *, / +, -	EFC	DGA	0:17
7	Is space hot?	FAC	BEG	0:19
8	How to teach division and inverse number in elementary school	BEG	FAC	0:35

Table 2. Topics and subjects (A-G) assignment for each chat session.

In each session, two groups discussed the same topic. One group used Reach, and the other group used Reach without letter-by-letter display function. We call this version of chat tool as “Normal chat tool”. After the discussion, participants evaluated (1) ease of conversation (2) choice of topic (3) choice of participants, for three scales from “good” to “bad”. Only the

ease of conversation showed difference between Reach and normal condition and Reach was more favored ($t(38)=2.77, p<.01$). Figure 7 shows average evaluation score in each topic. As for the choice of topic and choice of participants, they were not significant ($t(38)=0.24, p=0.81$ for topic and $t(38)=1.60, p=0.12$ for participants).

Fig. 7. Comparison of evaluation on ease of conversation in each topic

We also labelled Dialog act tags (DAs) to discussion log. They are: proposal / explanation / agreement / objection / question / answer / continuation / conclusion, and other. For the number of utterances (i.e. number of total DAs), no difference was found. However, the number of utterances might be dependent on topic ($F(6,13)=0.08, MSe=275.1, p=0.08$). Clearly, we have to gather more chat session log data to confirm this, because learners (subjects) pointed out the importance of topic choice. There was no difference about usage of DAs in two systems (Figure 8). Dialog acts such as “Explanation”, “continuation”, and “Other” are frequently used ($p<.01$ for Reach and $p<.001$ for normal system). There is no difference between Reach/Normal conditions or between subjects.

Fig. 8. Distribution of number of Dialog Act types in two chat tool conditions

We collected comments right after each session. Subjective comments revealed positive and negative aspects of the system:

- Typing process display can be used as credible awareness display.
- When I got accustomed to normal condition, I tended to less care of others’ typing. This was good for concentration, but turn taking tended to be selfish.

- In normal condition, discussion log tended to be cycle of question and answer. This was good for discussion log, but was awkward during the session.
- There is a danger of never using TALK command window, because everything can be seen in CHAT window.

4. Face mark Display of Dialog Acts

4.1 Estimation of Dialog Acts (DAs)

In the preliminary evaluation described in the previous section, learners preferred Reach to classic standard chat tool. Nevertheless, at the same time there was not significant difference of quantity of utterances and distribution of DAs between them. Therefore we thought we could enhance discussion by displaying DAs. By displaying DAs, we expected to visualize discussion flow and each learner’s contribution moderateley but not directly. We thought it would improve efficiency of discussion and coherency of the flow of logic.

Thus we implemented (1) module for detecting DAs and (2) displaying DAs. We used template matching method for identifying DAs because its implementation is simple and suitable for the small size of our data. It is usual that atmosphere of chat is casual compared to that of BBS. This means that variation of characteristic key phrases for each DAs is not too much wide, which would facillitate coding templates. We followed the method used by Boucouvalas (2003), where affect state in text chat is estimated in real time.

This new version of *Reach* identifies DAs by referring the templates illustrated in Table 3 just after a leaner sends a message to TALK window. Inventory of DAs is decided according to the past log of discussion (Figure 8). Among DAs in the log, “Continued utterance” and “Others” are excluded from because of its ambiguity. Thus we prepared seven types of DAs: explanation, question, proposal, conclusion, assertion, reply, agreement.

Template matching is applied from the top of sentence and every template is checked for the sentence. If multiple templates match a sentence, all the matched DAs are displayed. This is partly because of the imperfect precision of templates and partly because of our thought that DAs might not be mutually exclusive.

Dialog Act	Example of templates
explanation	because (<i>words</i>)
question	(<i>words</i>)?
proposal	how about (<i>words</i>)?
conclusion	in short, (<i>words</i>)?
assertion	I think (<i>words</i>)
reply	(yes no), (<i>words</i>)
agreement	(OK I see yes)

Table. 3. Templates for estimating Dialog Acts (DAs)

4.2 Display of DAs using face mark

Display of DAs can be done by adding some information to discussion log in the TALK window. However, directly showing DA tags might cause somewhat formal impression. We wanted to control degree of the effect of DA display so that we can keep learners relaxed and at the same time they can receive information about discussion status. In addition, we wanted the display simple not to increase cognitive burden of learners.

Thus we decided to use face mark for the Display of DAs. Face mark such as (^ o ^) (happy) is frequently used among Japanese young to middle age internet users and mobile phone users. The benefits of its use are that it is simple because it is text character and is easy to understand intuitively (Inoue 2006). We defined seven face marks for seven DAs (Table 4). Face marks are added to the corresponding sentence in Talk window. This time choice of the face marks are done by authors, however, we plan to find best suitable face marks hearing users in terms intuitive comprehension. A screenshot of the session is shown in Figure 9.

Dialog Act	Face mark	description of face mark
explanation	(' / ω ') < α ※ Ω 驟	saying something
question	(° へ °) ?	asking
proposal	(・ O ・) 」 < !	saying something important
conclusion	d (° o °) o	saying in determined attitude
assertion	ヽ 。 ・ ω ・ / < ヨロシク	vowing or asking
reply	(° ω °) / へい	answering in response
agreement	(° ー °) (。 ー 。) (° ー °) (。 ー 。)	nodding

Table. 4. Facemark for DA types and description of face mark representation

Fig. 9. A screenshot of face mark display and its translation to English.

5. Discussion

First we discuss result of evaluation. They are:

- (a) Display of typing process does not affect directly to number of utterances. These are more dependent on the types of topic.
- (b) Display of typing process can improve awareness and ease of conversation.

We think it is a positive point that number of dialog acts between reach and normal chat condition has no difference. This means that learners/participants use submission

function and build discussion history even though they can see all the typing process in private CHAT window. This may imply they use TALK window for past sentence reference. However, this must be more explicitly investigated and currently we are planning to set up new experiment to see this function is actually working.

As for the content of discussion, dialog act analysis does not show major difference between two conditions (Reach/Normal) by only counting numbers. However, it is interesting that some comments refer to the transition pattern difference of dialog acts. It might be worth looking more detail such as transition pattern (Markov transition model) for each discussion session. Our hypothesis was that smooth and comfortable discussion will enhance learning. We feel that at least we could improve the ease of conversation in text chat discussion. Assessment on enhancement of learning itself will be necessary as a next step.

Next, on the application of face mark for the display of DAs, one problem is the choice of estimation algorithm. Template matching is simple, however, definition of templates are burdensome and also arbitrary. We plan to apply machine learning approach for this identification problem. On the amount of information display, we also plan to display additional information such as ratio of DAs for each learner in CHAT window. This display would be either private or public, so that learner can control the level of disclosure. Evaluation of face mark display is not yet fully carried out. This will be the next step of our work. Comparison between this approach and explicit DA tagging by learners themselves would be also necessary.

The text character face marks have very simple looking compared to that used in Yahoo! messenger etc., where face marks are graphical icons. Because text characters are simple, their impression is moderate, and not intrusive. Interestingly, there is a traditional cultural tendency among Japanese that they love simple, unsophisticated expressions.

6. Summary

In this paper, we introduced a text chat tool that displays learners' typing process. The advanced feature of the system is the dual display mode: simultaneous display corresponding to non-verbal expression and classic text chat display that shows learners past history of dialog. We also developed tools for post analysis such as typing speed calculation or discussion visualization. Using this chat system, preliminary evaluation was carried out and revealed that the system can improve the ease of conversation. However, there was no essential difference of dialog acts between proposed system and usual text chat system. Learners' comments show both the merit and demerit for the proposed system. For example, it is good for improving awareness, but there is possibility not submitting completed sentences when they could see each other's typing letters.

We also added function of displaying DA face mark. This is to enhance logical discussion at the same time being remained in a relaxed mood. Efficiency of this function would be assessed in the next step.

Because so far we only focused on the conversation and did not assess how much participants actually learned during the discussion, we are now planning to apply this tool for more knowledge centric CSCL environment. We think we can find more detailed advantage and limitations of approach in realistic condition.

7. References

- Abrams, Z. I. (2003). The effect of synchronous and asynchronous CMC on oral performance in German. *The Modern Language Journal*, Vol. 87, No. 2, 157-167.
- Boucouvalas, A. C. (2003). Real Time Text-to-Emotion Engine for Expressive Internet Communications, In: *Being There: Concepts, effects and measurement of user presence in synthetic environments*, Riva, G.; Davide, F. & IJsselsteijn, W.A (Eds), pp. 306-318, Ios Press, Amsterdam.
- Cook, C.; Irwin, W. & Churcher, N. (2005). A user evaluation of synchronous collaborative software engineering tools, *Proceedings of 12th Asia Pacific Software Engineering Conference (ASPEC'05)*, pp. 705-710, Taipei, Taiwan.
- Duncan, S. & Fiske, D. W. (1977). *Face-to-face interaction: Research, methods and theory*, Hillsdale, NJ, USA, Lawrence Erlbaum Assoc Inc.
- Hayashi, S.; Yokoyama, S. & Akahori, K. (2004). Development and Evaluation of Synchronous Chat System for Discussion in E-learning, *Proceedings of International Conference on Computers in Education (ICCE2004)*, pp. 99-104.
- Inoue, S. (2006) *Netto Shakai no Wakamono Kotoba* (Words used in the Internet by younger generation, in Japanese), Gengo, Vol. 35, No. 3, pp. 60-67, Taishukan, Tokyo, Japan.
- ISO/IEC JTC1 SC36 WG2. (2004). Korea Standardization Activities for Collaborative Learning (N0074), <http://collab-tech.jtc1sc36.org/index.html>.
- Kitade, K. (2000). L2 Learners' Discourse and SLA Theories in CMC: Collaborative Interaction in Internet Chat, *Computer Assisted Language Learning*, Vol. 13, No. 2, pp. 143-166.
- Kienle, A. (2008). Intertwining synchronous and asynchronous communication to support collaborative learning—system design and evaluation. *Education and Information Technologies*. DOI 10.1007/s10639-008-9065-9.
- Koschman, T. (1996). *CSCL: Theory and Practice of an Emerging paradigm*. NJ: Lawrence Erlbaum Assoc Inc.
- Sacks, H. E.; Schegloff, E. & Jefferson, G. (1974). A simplest systematics for the organization of turn taking for conversation, *Language*, Vol. 50, pp. 696-735.
- Pittenauer, O. M. & Wagner, D. (2005) SubEthaEdit, <http://www.codingmonkeys.de/subethaedit>.
- Tamura, Y. (2005). Survey of Collaborative Learning Functions and Data Items on Commercial LMSs (in Japanese), *Proceedings of 30th Annual Conference of Japanese Society for Information and Systems in Education (JSiSE 2005)*, pp. 132-133, Kanazawa, Japan, August, 2005.
- Tsukahara, W., Ward, N. (2001). Responding to subtle, fleeting changes in the user's internal state, *Proceedings of ACM SIG-CHI on Human factors in computing systems (CHI 2001)*, pp. 77-84, Seattle, WA, USA, May 2001.
- Tsukahara, W.; Zisheng, Z.; Akamatsu, S.; Anma, F. & Okamoto, T. (2007a). Reach: an Awareness Rich CSCL Text Chat Display Tool with Real-time Key Typing, *Proceedings of The 7th IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, pp. 655-659, ISBN 978-0-7695-2916-5, Niigata, Japan, June 2007, IEEE Computer Society.

- Tsukahara, W.; Akamatsu, S.; Sakurada, K.; Wang X.; Ito, H. & Okamoto, T. (2007b). AN ATTEMPT TO ENHANCE CONVERSATION IN CSCL TEXT CHAT BY DISPLAYING LEARNERS'TYPING PROCESS, Proceedings of THE IASTED International Conference on WBE-BASED EDUCATON (WBE2007). pp. 568-571, ISBN 978-0-88986-650-8, Chamonix, France, March 2007, ACTA Press.
- Ward, N. (2004). Pragmatic Function of Prosodic Features in Non-Lexical Utterances, Proceedings of international conference for ISCA Special Interest Group on Speech Prosody (Speech Prosody 2004), pp. 325-328, Nara, Japan.
- Yamada, Y. & Takeuchi, Y (2005). A Consideration of Verbal Communication by the Freewheeling Chatting System, IPSJ SIG Technical Reports, Vol. 127, No. 2005 (2005-SLP-59), pp. 85-90, IPSJ.

IntechOpen

Advances in Learning Processes

Edited by Mary Beth Rosson

ISBN 978-953-7619-56-5

Hard cover, 284 pages

Publisher InTech

Published online 01, January, 2010

Published in print edition January, 2010

Readers will find several papers that address high-level issues in the use of technology in education, for example architecture and design frameworks for building online education materials or tools. Several other chapters report novel approaches to intelligent tutors or adaptive systems in educational settings. A number of chapters consider many roles for social computing in education, from simple computer-mediated communication support to more extensive community-building frameworks and tools. Finally, several chapters report state-of-the-art results in tools that can be used to assist educators in critical tasks such as content presentation and grading.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wataru Tsukahara, Shingo Akamatsu, Xin Wan and Toshio Okamoto (2010). Realization of Swift and Expressive Chat in CSCL: Reach Project, *Advances in Learning Processes*, Mary Beth Rosson (Ed.), ISBN: 978-953-7619-56-5, InTech, Available from: <http://www.intechopen.com/books/advances-in-learning-processes/realization-of-swift-and-expressive-chat-in-cscl-reach-project>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen