A Documentation Platform for Supporting and Assessing Collaborative Knowledge Building in Learning Computer Programming

SUN Zhi, LI Zhe, ZAORSKI Spence, NISHIMORI Toshihisa, MAESAKO Takanori, NAKAMURA Masako, IMAMURA Rie

【要旨】

本研究では、知識構築を促進するため、ドキュメンテーションを通して学習者に既定の学習内容に沿って学習を進めさせ、教授者との形成的なコミュニケーションを深めるとともに、学習者一人ひとり、及び協同学習によるグループの学習過程や学習成果を記録した持続的なポートフォリオを提供する方法を考案した。この方法を用いて、実践的なプログラミング能力の習得を目的とする教育実践を行い、その効果を検証した。学習者によりよく学習内容に集中して学習活動を行わせるために、教育実践を設計するための理論的枠組みを設定し、また、この理論的枠組に基づいてプログラミング学習のプラットフォームを構築した。複数の学習者が同時にこのプラットフォームにアクセスして、オンラインでドキュメンテーションを共有し作業することができる。また、ほかの学習者が作成したものを閲覧したり、コメントしたりすることもできる。その結果、学習過程や学習成果の記録や、協同学習を通してプログラミングへの理解が進んだ。以上より、協同的なドキュメンテーション作業は知識構築を促すことに有効であることが分かった。

1. Introduction

In a knowledge society, the greatest challenge to education is not how to effectively help learners to acquire a defined set of knowledge and skills, but in helping them to learn how to manage and work creatively with the creation of new knowledge (Law & Wong, 2003). Knowledge building, defined as “the production and continual improvement of ideas to a community” (Scardamalia & Bereiter, 2003), emphasizes the improvement of ideas and collaboration in a community. Education focuses on learning for which the goal is to enhance personal knowledge, but in the knowledge building approach to education, the focus shifts to the construction and advancement of collective knowledge (Lamon, et al., 2001), and to interactive and collaborative learning from individual learning in a non-contextual situation (Gan, 2005). It can be concluded that most of the research in knowledge building has formulated a perspective on knowledge building as a social process of collaborative learning.

However, in a classroom, this approach also raises several practical pedagogical questions due to its collaborative effect. Knowledge building requires students to take over a significant portion of the responsibility for their own learning, including planning, execution, and evaluation (Bereiter & Scardamalia, 1993). Collaborative knowledge building driven by students themselves usually leads to less exposure of personal work and social dynamics during task completion process (Lee & Lim, 2012). Also, because it is common for students to choose to divide tasks into parts for the members of a group, it also might produce some knowledge blind spots in the parts undertaken by others (Sun, et al, 2014). This can result in social loafing, the tendency to reduce individual effort when working in groups compared to the individual effort expended when working alone (Williams & Karau, 1991). Considering that personal learning is the preparation for collaborative knowledge building, it is necessary to pay attention not only to collective artifacts creation, but also to individual work that contributes to community knowledge.

To address these issues, this paper presents a documentation approach to promoting knowledge building on the premise that the documentation process would develop knowledge building portfolios that give access to the work leading up to completion and provide a persistent record of individual and collective work. Utilizing this documentation approach, we created an instructional design model structuring students' learning activities in two dimensions: theoretical/practical and personal/collaborative. An online collaborative documentation platform custom built to support this model was developed to facilitate students' personal and collaborative documentation work. The documents monitored and tracked using this platform served as the expression of students' understanding of learning content and the recordings of the learning process were used to analyze students' learning.

2. Documentation approach

Documentation initially introduced in education served as the learning process portfolio recorded by teachers or other observers. Katz and Chard (1996) stated that "documentation typically includes samples of a child's work at different stages of completion; photographs showing work in progress; comments written by the teacher; transcriptions of children's discussions, comments and explanations of intentions about the activity". The educators' conception of documentation as combining many forms of texts makes learning visible. Documentation as collective knowledge building artifacts is widely used in computer science, especially in software documentation. In "The American Heritage Dictionary of the English Language", the term of "documentation" is defined as "the organized collection of records that describe the operation and use of a program, operating system, or hardware device". This documentation developers' conception is aimed at sharing and improving expert knowledge and experience for the developer community.

From these definitions, we can summarize several characteristics of documentation. First, the documents in documentation will provide a persistent record of experiences, discussions, problems, as well as solutions to the problems. In computer science, they serve as a reference for other developers, users and learners. Then the documentation is kept continuously improved by collaboratively sharing, discussing and reaching mutual insights. And for the documentation writer himself, the documentation process helps him develop a deep understanding by giving a detailed description of contents.

This research develops a documentation approach that involves students in a series of documentation tasks conducted individually and collaboratively in order to visualize their understanding of learning contents and their learning process. The distributed documentation work will increase the control and external drives on these students with less self control. The collaborative documentation work engages students in making collective inquiries regarding personal documentation work and realizing deeper comprehension through self-expression, interactive questioning, arguing, and coming to agreement. The documents produced during the documentation work serve as a persistent record of individual and collective artifacts. The documentation database also serves as learning reference for other students as well as for teachers.

3. Course implementation

The course, "Seminar in Educational Technology" was implemented according to course syllabus shown below (Fig.1). The class was composed of 11 third-year undergraduate students studying Educational Technology and preparing for their course assignment of "developing a digital textbook for undergraduate students" using HTML, CSS, and JavaScript. Computer programming was chosen as the learning content for this study and we conducted two programming lessons in June 2014.

Course Syllabus (Seminar in Educational Technology)			
4/09	Guidance regarding digital textbook	6/04	Preparation for digitalization
4/16	Proposal for textbook	6/11	Basic programming knowledge
4/30	Preliminary textbook	6/18	Programming learning
5/07	Presentation of textbook	6/25	Programming learning
5/14	Improvement of textbook	7/02	Presentation of artifacts
5/21	Prototyping textbook	7/09	Prototyping artifacts
5/28	Summarization on prototyping	7/16	Evaluation and summarization

Fig.1 Course syllabus

3.1 Instructional design for documentation work

Traditional computer programming lessons based on lectures and practical laboratory work focus on reproducing the program that is being taught (A. Robins, et al, 2003). Our interest in this field is focused on supporting students learning computer programming through documentation work. Students were organized to do a series of documentation tasks building up to the production of a programming document based on the students' own learning experiences. With practical and collaborative considerations, we carried out instructional design for the computer programming lessons to structure students' learning activities using documentation work (Fig.2).

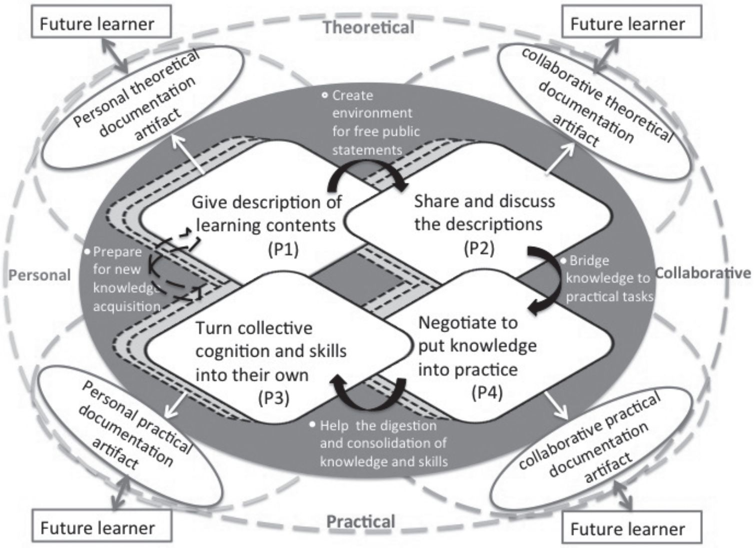

Fig.2 Instructional design for documentation work

The whole process of knowledge building can be seen as a continuous upward spiral with previous knowledge serving as preparation for new knowledge acquisition. Learning activities in each section of the spiral are structured in four parts with two dimensions: theoretical/practical and personal/collaborative. In Fig.2 above, the diamonds represent the students' learning activities and the arrows represent the teacher's supporting scaffolds. Under the teacher's help of building bridges between former knowledge and new knowledge, the learning process begins with phase1 (P1: personal learning phase), shown as the upper-left quadrant in Fig1, with consideration of the importance of personal preparation for more effective collaboration. In this work, the documentation work in P1 engages students in documenting what a particular attributes or functions do in several example programs. Next phase2 (P2: collaborative learning phase) involves making a collective inquiry regarding personal documentation work and realizing deeper comprehension through self-expression, interactive questioning, argument, coming to agreement, and developing collaborative documentation work. In phase3 (P3: collaborative practice phase), students conduct several practical tasks (in this work, programming task of collaboratively developing a new program, including program design, compiling, debugging, with the aim of putting these theoretical knowledge into practice). Finally, students digest and absorb the collective knowledge and skills for themselves in phase4 (P4: personal practice phase). In this work, students were engaged in "the production of a digital textbook for undergraduate students" as a personal artifact. Phase4 can also be seen as the preparation for the next cycle of the knowledge building process. We also bring in an external factor in this model, future learners as the target users of students' artifacts, to improve students' motivation. Learners in the future are assumed to experience and refer to students' artifacts produced in each phase.

Two programming lessons were conducted based on this instructional design. The instruction workflow (Table.1) shows that students learned from instructional videos individually to get started with basic programming knowledge (variables, attribute, function, etc.). In the first lesson, students were organized into groups and each group of students was given a different programming technique to learn, first as individuals when they did documentation for the program learned in video, then as a group when they discussed the documentation work together. Then in the second lesson, students were shuffled into different groups and given a shared activity. Each student first presented his/her documentation work of programing techniques to the group and then worked on a single group program by using their collective experience from studying the different techniques in their previous groups. Since class time was limited, students were assigned to watch the online instructional videos before class and to work on their individual digital textbooks after each lesson so that more time could be used interacting with other students.

Table.1 Instruction workflow

Before class:
Learn from instructional videos individually
First programming lesson: Split into 4 groups (11 students) with different learning contents for each group
P1: Do documentation for program (Personal learning)
P2: Discuss and revise the documentation work (Collaborative learning)
Second programming lesson: Split into 3 groups (9 students) with different learning contents for each group member
P2: Discuss and revise the documentation work (Collaborative learning)
P3: Revise the documentation work (Collaborative practice)
After class
P4: Develop personal digital textbook (Personal practice)

3.2 Design and development of documentation platform

To support students' documentation work for learning programming, the premier focus of our work was the development of a collaborative programming learning platform based on the instructional design for documentation work. Students worked on the platform as they practiced programming techniques learned in class, consulted resources to better understand the programming techniques, revised their documentation work, and referred to other students' work to improve their own understanding. The platform serves as a shared database (Fig.3) of programs that the teacher and students produced. It is a means of referencing and commenting on one another's contributions.

Fig.3 The programs database

The platform was used to help students develop their own programs. HTML, CSS, JavaScript and Output windows were viewable side-by-side (Fig.4), making program experimenting, attributes or values changing, and program debugging more intuitive. It also provides Automatic error detection to check students' work. To support students' collaborative work, the platform gave students simultaneous access to a collaborative programming environment. Student could share the programming interface that they were working on by inviting others to view, check, and comment on their files, as well as to edit a program collaboratively in real time.

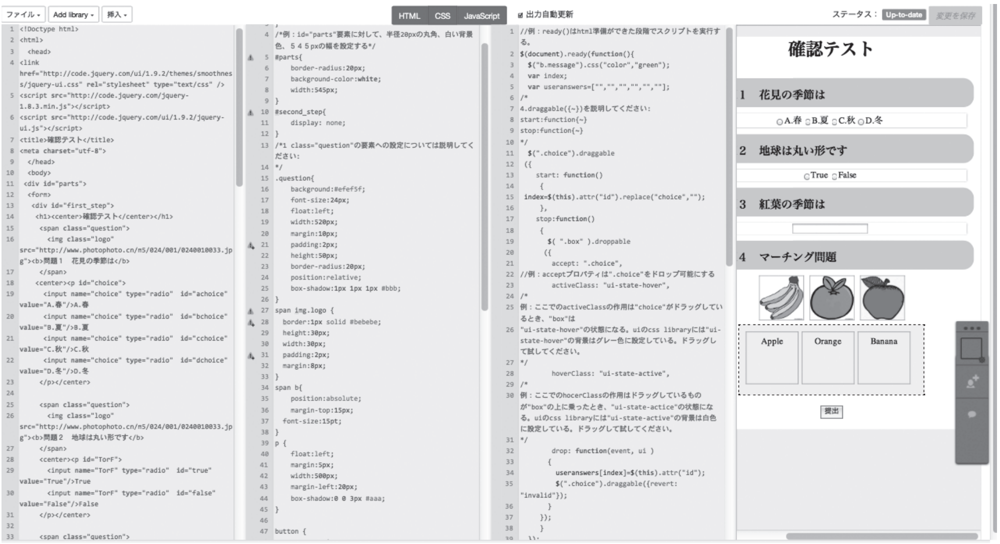

Fig.4 Programming environment

The platform records students learning experiences, discussions, and problems; it does not only show the final work, but also the work process leading up to artifact creation. In Fig.5 below, the left part is a former version and the right part is a later revision, revealing a student's programming process by comparison of the two. Detailed records of student access and activities serve as learning process portfolios and give the teacher more opportunities to provide support and feedback.

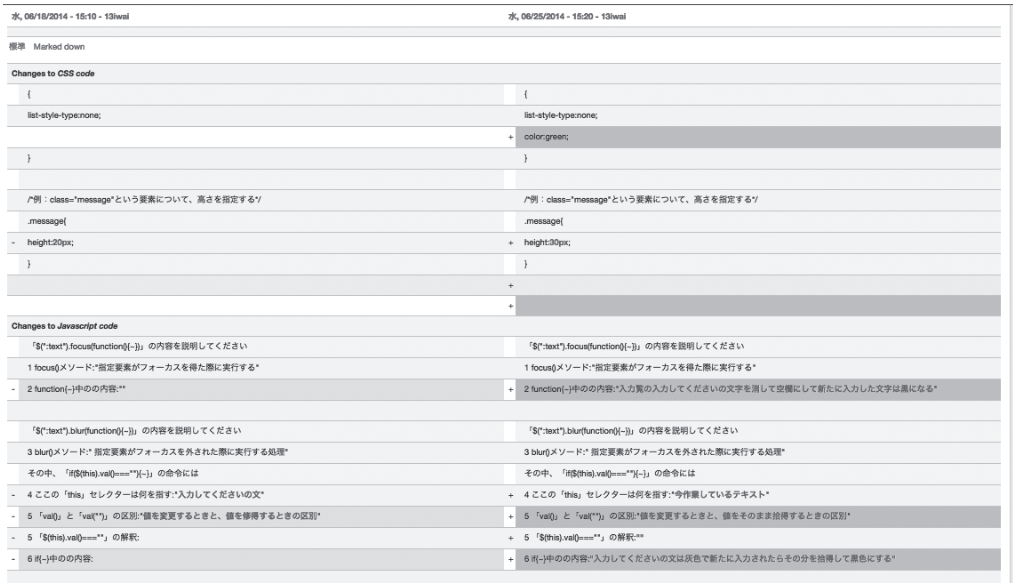

Fig.5 The comparison of different edition of students' work

To optimize personal knowledge acquisition, students could access the learning materials in the platform anytime from anywhere. We prepared screencast videos as online learning materials that could be used in place of traditional lectures. Other learning materials also included the example programs, a reference sheet, and other related materials. The platform was accessible over the Internet using standard web browser software so students could access it during class as well as at home.

3.3 Data sources

Documentation works

The documentation work done in different phases served as knowledge building portfolios and we can learn the knowledge building process by comparing different versions in different learning phases. Students needed to write explanatory statements for programming techniques in their documentation works. The explanatory statements in their documents were coded using a 5-point scale (1-no explanation, 2-wrong explanation, 3-unclear explanation, 4-right explanation, 5-very clear explanation).

Questionnaire

Students were asked to complete a questionnaire in which they provided subjective evidence for the collaboration-related knowledge-building principles (Scardamalia, 2002). Seven items (Table.2) of the 24 Likert scale questions from this attitude survey questionnaire were used for analyzing the relation between documentation work and collaboration-related knowledge-building principles.

Table.2 The items extracted from questionnaire

Q1	Develop basic understanding from instructional videos	1 2 3 4 5
Q7	Understand programming techniques by documentation work	1 2 3 4 5
Q8	Share documentation work actively	1 2 3 4 5
Q11	Pose problems for further discussion	1 2 3 4 5
Q13	Endeavor to answer questions posed by others	1 2 3 4 5
Q15	Become aware of something new by sharing	1 2 3 4 5
Q16	Get better understanding by discussing with other members	1 2 3 4 5

4. Results

4.1 Documentation works

Through analysis of the data gathered from students' personal and collaborative documentation work, we were able to gain insight into students' learning processes (Fig.6). The recording of work leading up to the artifact creation shows the improvement of understanding of model programs, which peaked during the collaborative work. In order to gain insight into the performance of students in the collaborative learning process, we analyzed each student's scores for each stage, shown in Table.2. Students A, H, J and K, whose scores were constantly on the rise, can be seen as the contributor role in their groups (marked as "○" in the table). Student B, F, G and I, experiencing a fluctuation in post-tests, played the dependent role in their groups (marked as "△" in the table). Student C was absent in the post-test and student D and E were absent in the second collaboration so missing values are marked as "□" in the table.

From Fig.6 and Table.3, we learned that except student K who had high scores from the beginning, most of the students started learning programming as novices. Student A achieved the most significant continuous progress. Although there was some progress in collaborative work, student C did not participate in collaboration actively, reflected in scores for collaborative work lower than other members in both two groups. Student F achieved a score for collaborative work similar to members in the same group but didn't score well as an individual. This means that student F's collaborative work may depend on the other group members, most likely student D and E, who were absent in the second collaboration. Student H played an important part in both collaborative works, because the other members in the same group (students G and I the first time and student B the second time) achieved similar performance but did not maintain the result to the end. Student J's performance had some deviation from other group members but kept steady growth.

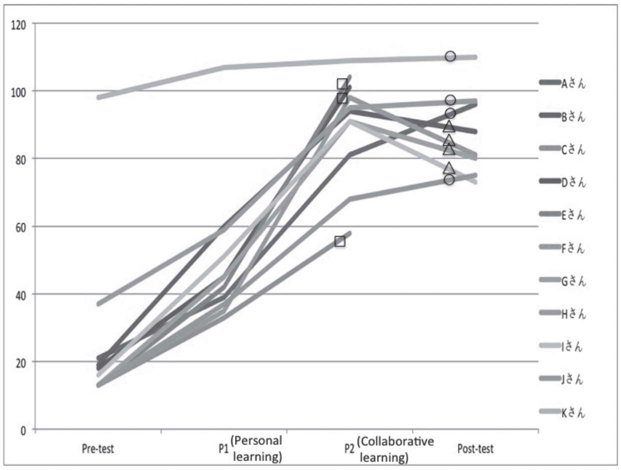

Fig.6 Scores attained in different stage

Table.3 Analysis of students' performance in group work

Phase 2	Group1	Group2	Group3	Group4
Collaboration first time	A B C ○△□	D E F □□△	G H I △○△	J K ○○
Collaboration second time	A F G ○△△	B H J △○○	C I K □△○	D E (Absent) □□

The platform also recorded students' learning experiences and artifacts in other learning phases. The group practical work artifacts produced in Phase 3 (Collaborative practice) showed that they were not prepared for developing a new program in such a short time. In their personal digital textbooks produced in Phase 4 Personal practice, students used most of the programming techniques that they learned in their previous groups. Students' browsing history showed that all of the students watched videos before class, but only one student watched an instructional video more than twice so it is necessary to take some measures to ensure efficient use of learning resources in the future.

4.2 Questionnaires

A Principal component analysis with a rotation method of Promax with Kaiser Normalization from this attitude survey questionnaire was conducted on data gathered from 9 participants, and the results are shown in Table.4.

Table.4 Rotated component loadings for 7 survey items*

Component	1	2
Q1 Develop basic understanding by online instructional videos	.483	.394
Q15 Be aware of something new by sharing	.782	-.449
Q7 Understand programming techniques by documentation work	.917	-.112
Q16 Get better understanding by discussing with other members	.943	.186
Q11 Pose problems for further discussion	-.128	.876
Q8 Share documentation work actively		.922
Q13 Endeavor to answer questions posed by others	-.125	.985
Eigenvalues	2.998	2.551
Percentage of variance	42.824	36.441
Number of test measures	4	3

*Loadings =>.10

When loadings less than 0.50 were excluded, the analysis yielded a two-factor solution with a simple structure (factor loadings =>.50). It is clear from Table.4 that these four items loaded onto Factor1 all relate to an improvement of ideas and understanding, so this factor was labeled as "Rise-above". Three items load onto a second factor related to the students' contribution to their groups in collaboration. Factor2 was labeled as "Collective responsibility". The label for factors is based on the Collaboration-related knowledge building principles (Scardamalia, 2002).

Liner regression analyses were conducted to examine the relationship between the two factors with "Role in the group", coded as 1= dependent role, 2= contributor role, 9999= missing values (absent). Table.5 summarizes the descriptive statistics and analysis results. As can be seen, Factor2 "Collaborative responsibility" is positively and significantly correlated with the "Role in the group", indicating that those with higher scores on collaborative responsibility tend to be the contributor role in collaboration. Students with an improving score were expected to have more collaborative responsibility in-group. The Liner regression model with all two predictors produced $R^2 = .865^*$, $F(2, 5) = 16.71$, $.01 < p < .05$. Factor2 "Rise-above" did not contribute to the liner regression model.

Table.5 Summary correlations and results from the regression analysis

Variables	Correlation with Score process	Standardized coefficients β
Factor1 Rise-above	.061	.127
Factor2 Collaborative responsibility	.922**	.931*
Adjusted R square	n=8	R^2 = .865**

* p < .05, ** p < .01, ***p<.001

5. Future work

Through analysis of the data gathered during our collaborative documentation project, we were able to gain insight into the learning processes of students. Analysis of this data showed how students build knowledge over the course of different phases. For instance, we saw that students developed a better understanding after the collaborative demonstration work in Phase 2. The collaborative nature helped to expose student learning and gain better understanding during their knowledge building process. Computer programming is very complicated and it may have been difficult if not impossible for a single student to create a usable digital textbook in such a short period of time. In future studies, we would like to set a less technical task for the end of the course and further explore patterns of student learning during documentation work.

References

Ahn, D. & Syngjoo, C. 2007 “Electronic portfolios: Blending technology, accountability and assessment”, *T.H.E. Journal*, 31(9), pp.12–18.

Barron, et.al. 1998 “Doing With Understanding: Lessons From Research on Problem-Based and Project-Based Learning”. *Journal of the Learning Sciences,* vol. 7(3&4), pp. 271–311.

Frank, M. & Barzilai, A. 2004 “Integrating alternative assessment in a project-based learning course for pre-service science and technology teachers”. *Assessment & Evaluation in Higher Education*. 29(1), pp.41–61.

Gan, Y.C. 2005 “Knowledge Building and Collective Wisdom Advancement in Virtual learning Communities: Perspective on the Integration of Knowledge Management and e-Learning”, Beijing, China: *China Educational Science Publishing House*.

Gan, Y., & Zhu, Z. 2007 “A Learning Framework for Knowledge Building and Collective Wisdom Advancement in Virtual Learning Communities”. Educational Technology & Society, 10(1), pp.206–226.

Lamon, M., Reeve, R., & Scardamalia, M. 2001 “Mapping Learning and the Growth of Knowledge in a Knowledge Building Community”. *Paper presented at American Educational Research Association Meeting 2001*. Seattle, Washington, retrieved December 7, 2006, from http://kf.oise.utoronto.ca/lamon/mapping.html.

Law, N., & Wong, E. 2003 “Developmental trajectory in knowledge building: An investigation”. In Wasson, B., Ludvigsen, S., & Hoppe, U. (Eds.), *Designing for Changes, Dordrecht: Kluwer Academic Publishers,* pp.57–66.

Muir, S.P. & Tracy, D.M. 1999 “Collaborative Essay Testing: Just try it!”, *The Journal of College Teaching*. 47(1), pp.33–35.

Robins, A., Rountree, J. & Rountree, N., 2003 “Learning and Teaching Programming: A Review and Discussion”, *Computer Science Education,* 13(2), pp.137–172.

Romano, N.C. & Nunamaker, J.F. 1998 “A project-centered course: Collaborative computing”. *Proceedings of HICSS-31,* New York: IEEE.

Scardamalia, M., & Bereiter, C. 1996 “Computer support for knowledge-building communities”. In Koschmann, T.

(Ed.), *CSCL: Theory and practice of an emerging paradigm,* Hillsdale, NJ: Lawrence Erlbaum, pp.249–268.

Scardamalia, M. 2002 “Collective Cognitive Responsibility for the Advancement of Knowledge”, In: B. Smith (ed.), *Liberal Education in a Knowledge Society*. Chicago: Open Court, pp.67–98.

Scardamalia, M., & Bereiter, C. 2003 “Knowledge Building”. In Guthrie, J.W. (Ed.), *New York: Macmillan Reference,* pp.1370–1373.

Stahl, G. 2002 “Contributions to a Theoretical Framework for CSCL”, *in Proceedings of the Conference on Computer Support for Collaborative Learning: Foundations for a CSCL Community,* Boulder, Colorado, pp.62–71.

Solomon, G. 2003 “Project-based learning: A primer”, *Technology & Learning*. 23(6), pp.20–26.

Sun, Z., Zaorski, S., Nakamura, M., Okubayashi, T. & Maesako, T. 2014 “Using Collaborative Documentation to Assess the Learning Process During Project-based Learning”. In M. Searson & M. Ochoa (Eds.), *Proceedings of Society for Information Technology & Teacher Education International Conference 2014*. Chesapeake, VA: AACE. pp.1456–1461.

Tynjala, P. 1999 “Towards expert knowledge? A comparison between a constructivist and a traditional learning environment in the university”, Int. J. Educ. Res.

A Documentation Platform for Supporting and Assessing Collaborative Knowledge Building in Learning Computer Programming

SUN Zhi, LI Zhe, ZAORSKI Spence, NISHIMORI Toshihisa,
MAESAKO Takanori, NAKAMURA Masako, IMAMURA Rie

Abstract This paper presents a documentation approach for promoting students' knowledge building in learning Computer Programming. The underlying premise is that the documentation process allows students to track their learning, develops transformative communication with teachers, and provides a persistent record of individual work and collective cognition artifacts. Our focus is on practical and collaborative considerations for learning computer programming, and we explore a theoretical instructional design framework that organizes learning activities in order to better engage students in the programming course. A learning platform to support this theoretical framework was developed to give students simultaneous access to a shared online documentation work environment. The students were able to program and do documentation work together in real time, and to invite others to view, check, and comment on their files. The documentation work leading up to the artifact creation showed the students' improving understanding of programming, especially in their collaborative work. Thus, we show how collaborative documentation work is useful in improving deep constructivism and students' engagement in knowledge building.