Realization of a collaboration system for everyone to develop and manage

— Practices of communication patterns using qwikWeb —

Kouichirou Eto [*], Masahiro Hamasaki [**] and Takuichi Nishimura [**]

[Translation from *Synthesiology*, Vol.1, No.2, p.101-110 (2008)]

To realize a new collaborative communication system, we propose qwikWeb which has a design philosophy based on communication patterns where a user can easily build a system that matches the group's activity. Adequacy and efficacy of this system were demonstrated by designing, implementing, operating, improving, and conducting analysis of operation data.

Keywords : Collaboration system, groupware, pattern language, Wiki, mailing list

1 Introduction

With the advance of communication and Internet technology, we now possess the means for sharing information over time and space. As of 2005, there were estimated 69.23 million users of portable terminals such as cell phones, and estimated 66.01 million people used personal computers[1]. If we can provide a system that can be built collaboratively within an organization, organizations, interest groups, and local areas can more actively engage in creative and economic activities. However, currently available systems were difficult to manage, and users could not efficiently accumulate and structure knowledge.

In this paper, we propose qwikWeb[2], a collaboration system for user groups who wish to have a system that does not require complex access control, is easy to manage and learn, and enables accumulation and structuring of knowledge, as shown in Table 1. Communication patterns were employed as background design philosophy that enables construction of a system to a user's preference. Adequacy and efficacy of the system were demonstrated by designing, implementing, operating, and improving the system, and conducting analysis of the operation data.

In this paper, Full Research of the web system is discussed in Chapter 2. Corresponding to this development process, present status analysis of the collaboration system is described in Chapter 3. Service design of qwikWeb is discussed in Chapter 4, and implementation of qwikWeb in Chapter 5. Data analysis and discussion of qwikWeb in operation are discussed in Chapter 6. Related studies are discussed in Chapter 7 to clarify the positioning of this study, and summary and future directions are presented in Chapter 8.

Table. 1 Comparison of existing collaboration systems and qwikWeb.

	Ease of management	Accumulation and structuring of knowledge	Complex access control
Cybozu[3] and others	×	△	○
qwikWeb	○	○	×

2 Full Research of the web system

In the Full Research for developing and realizing the web system, it is important to develop the system by understanding user behavior and by utilizing user feedback. The basic design and development procedure involves the participation of the users as well as the designers and the system developers, as shown in Figure 1.

First, in the phase for analyzing the current situation, outline of service and target users are selected, and current problems are analyzed. Since service outline and target users

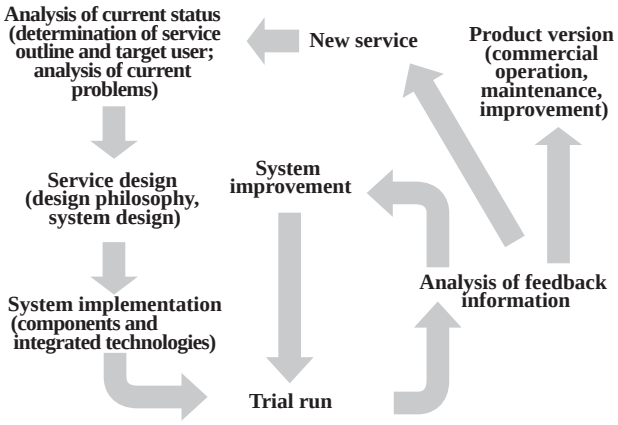


Fig. 1 Development process of web system.

* Center for Service Research, AIST ** Information Technology Research Institute, AIST 2-41-16, Aomi Koto-ku, Tokyo 135-0064, Japan * E-mail: k-eto@aist.go.jp

can be identified by understanding the current problems, analysis of the current situation can be done simultaneously. Extraction of problems in the current system is done on a concept level as well as on a functional level. New service is designed based on these results. In this design, design philosophy including service vision is determined to create a system design incorporating selection and improvement of component as well as future scalability and linkage with other systems. System implementation follows system design, and cost performance in terms of software engineering is taken into account. Next, test run is done with a certain number of users. In this phase, not only is it necessary to accumulate system log to collect user feedbacks, but it is also necessary to hear direct user comments. Gathered information is analyzed and the design is improved as needed, and the commercial version is completed as the system is continuously improved on the test run. During the analysis and improvement processes, hidden demands may become uncovered and findings that enable implementation of new services can be collected. The commercial version will be sold or licensed to a management company for commercial operation, but maintenance and system improvement must be continued throughout the operation.

Described above is the user participation process in Full Research for the web system. qwikWeb was developed according to this process, and currently, the test operation is almost complete. The development process of qwikWeb will be discussed in the following chapters.

3 Analysis of the current state of the collaboration system

In cooperative and collaborative work in organizations such as companies and interest groups, e-mail plays an important role, and communication and collaboration using the mailing list (ML) is done very actively. However, in collaborative work, information may not be structured, and it requires time to confirm the latest information in many cases. On the other hand, collaboration tools or groupware such as Lotus Notes[3] and Cybozu[4] are used since they have functions for schedule adjustment, table making, and collaborative document creation. However, set-up procedure and operation of such systems are complicated, and the managers are heavily burdened by maintenance work. Also, since communication method within the groupware is different from e-mail address used regularly, there is extra work of data transfer when building document and knowledge collaboratively by e-mail exchange. Flexible cooperation with members outside the group is not easy.

For a user who must keep the costs of start-up and operation low, and wishes to have a collaboration system that does not place a burden on certain managers, it is necessary that he/she can understand how to use the system just by looking at it, without extensive training on the system. Also, the system must have expandable structure that enables gradual construction of user's ideal information environment, built up from the environment with which the user is already familiar. For such a user, detailed access control (for example, which user can edit which document) for each document and knowledge is unnecessary. This is because, while systems like Cybozu[4] assume several thousand users, a user who cannot rely on a manager needs a collaboration system for just a few dozen people.

As an example of target users, the Authors looked at collaborative work by researchers who could not spare time for groupware management. The researchers used e-mail regularly as a main mode of communication. However, they did not spontaneously use groupware in collaborative work unless ordered by the supervisor to create documents. This was also true even in a situation where it was much more efficient to do collaborative document writing. Why was this? Okada[5] positioned communication at directly below collaboration in his hierarchical model of cooperation to emphasize the importance of communication. Kitagawa[6] divided the online community by three functions – obtain information, establish relationship, and collaborate – and stated that although majority of the communities existed simply to obtain information, the members became close in some communities, and some of them developed into collaborative communities. Close communication is necessary as background of collaboration, and collaboration is only possible through information exchange and agreement through communication. In fact, although collaborations such as report write-ups are done over e-mail, there is a problem in terms of structuring and sharing knowledge because all mails must be read and everything must be corrected and integrated to comprehend the latest information. Therefore, we saw a demand for a system that enabled structuring of intra-group knowledge and writing of collaborative document based on natural intra-group communication flow.

The issues were "easy management," "easy learning," and "accumulation and structuring of knowledge."

4 qwikWeb service design

4.1 qwikWeb design philosophy

4.1.1 Basic philosophy

As a result of considering the collaboration system issues mentioned in Chapter 3 from the standpoint of users, we set the basic philosophy of qwikWeb as: "users can freely design the system and the ways it is used."

Since it was not possible for general users to alter the system in Lotus Notes[3] and Cybozu[4], it was difficult to incorporate this basic philosophy. Therefore, we studied Wiki, which was developed by Ward Cunningham, and found it was designed

with a philosophy that closely resembled our proposal. We call Wiki's design philosophy "communication patterns"[7] [8], and we present it in the next chapter, and also discuss the design philosophy of qwikWeb.

4.1.2 "Communication patterns" - Wiki's design philosophy

The philosophy of pattern language[9] proposed by architect Christopher Alexander is reflected in Wiki. Pattern language is a summary of language-like expression of architectural forms that appear repeatedly in architecture. This enables the user of buildings to participate in the design process, which is expected to result in good architecture. In 1987, Wiki developer Ward Cunningham became interested in pattern language, and tried to apply it to software from the perspective of user participation in design. Several patterns were extracted as user interface, and it became possible to design excellent software in relatively short time by providing these patterns to users[10]. This experiment developed into a large movement for creating design patterns that served as common language among developers[11]. Cunningham used HyperCard launched in 1987 from Apple Inc., and created the browser to record and edit the discovered patterns[12]. This pattern browser later grew into Wiki. Wiki was originally developed as a base for recording and editing pattern language.

In 1995, Cunningham recreated the pattern browser as a system on web based on the contents accumulated so far, and called it WikiWikiWeb. This was the origin of Wiki. The greatest difference between the pattern browser on HyperCard and Wiki was the real time connection to the Internet. Wiki was no longer merely a place to accumulate information, but also became a place to discuss accumulated information.

In general, a system used for communication is managed by organizing information into some uniform format. A typical example is the bulletin board. An article is stored by breaking it down into several items such as author's name, date, document title, and text. Authorization is attached to the article, and usually only the author can edit the article. It is like sharing a piece of paper in contrast to the storage of information in Wiki. In Wiki, there was no set format of information and any document could be written freely. Anyone could edit any document since there was no concept of authorization. However, when several authors edited the same documents, confusion occurred inevitably. Local rules were created to solve such confusion. For example, "one must leave one's name at the end of a comment." Gradually rules were created within Wiki, which was in a state of anarchy in the beginning. Rules of communication developed spontaneously among user relationships, and they seemed to have a similar structure to the development of patterns in architecture. Therefore, in this paper, rules of communication that arise in this manner are called "communication patterns."

Although the initial objective of Wiki was to collect patterns that were necessary to create information systems, it soon became a place of communication among people who were interested in patterns, a place to discuss the rules (communication patterns) of communication, and also a place to record and collect communication patterns that were generated spontaneously. Wiki developed as a place not merely for collecting patterns but a place with a higher-level function of collecting the rules for collecting information (communication patterns).

The most important communication pattern in Wiki can be categorized into a thread mode and a document mode. The thread mode is a mode in which discussion is in progress using that page, and the user comments are presented in bulletin board form. In contrast, the document mode is a mode where only objective descriptions are presented on the page and subjective comments are excluded. In practice, most pages use a hybrid mode where objective descriptions are presented at the top of page and a discussion goes on in the lower part. Each page starts with a short document that describes the topic of the page, and comments on this topic are added in the thread mode. Individual differences are gradually absorbed as discussion progresses, and finally it grows into one objective statement (the document mode). It was expected that all pages would ultimately reach the document mode.

Here, rather than setting rules and communicating along the set rules, the stance is emphasizing spontaneous rulemaking where the rules on how to carry on communication are decided through communication. This stance is the most distinct characteristic of Wiki. If the essence of Wiki's design philosophy is described in one phrase, it is "emphasis on communication patterns."

4.1.3 Communication patterns in qwikWeb

Investigating the design philosophy of qwikWeb based on Wiki's design philosophy "communication patterns" which was restated in the previous section, it is a "system that enables the user to construct collaboration system appropriate for the group by combining various communication patterns." Adding the user's standpoint that it "can be used with current knowledge only, without explanation," it is a web system where one can practice expandable communication pattern to gradually develop one's ideal information environment, starting from a communication environment of e-mail that is very familiar.

4.2 System design of qwikWeb

4.2.1 Basic design

Considering "easy management," "easy learning," and

"accumulation and structuring of knowledge" that were discussed above, we investigated the routine communication method used by our target users. They were mainly telephone, facsimile, cell phone, e-mail, and web browsing. Among these, e-mail and web browsing were communication means of digital information such as text information which is compatible for distribution on the Internet. Since Wiki, whose design philosophy was adopted as described in the previous section, was an extension of web browsing and did not alter the way of using e-mail, it was assumed that a user would use mail in a conventional manner. Therefore, e-mail and Wiki were combined for structuring knowledge and for collaborative work. It was designed as a web system that enabled communication patterns in which the user could freely edit and set convenient rules including the image of the entry page to Wiki. Specifically, the following points were emphasized in the design.

- Ease of management
  It can be set up by sending e-mail, and then responding to the mail from the system. Anyone can add or delete members, thereby enabling collaboration with members outside the organization, which was difficult to do with the conventional system. Members of the system and Wiki can be matched so anyone can easily take on managerial work.
- Ease of learning
  Starting with simple communication using e-mail, one can gradually build up a desired information environment.
- Accumulation and structuring of knowledge
  Exchanges of e-mail are accumulated in Wiki, and necessary knowledge structuring and collaboration can be accomplished. Notification about who did what in Wiki is transmitted (distributed by mail).

Therefore Wiki and ML that distributed mails to the group were combined. ML is a mechanism whereby e-mail can be sent to several people simultaneously. When a person sends e-mail to the ML, it is then sent to all registered members. ML can be categorized as an asynchronous communication system. There are many server softwares for ML such as fml and majordomo, but a hosting service such as Yahoo Group[13] and freeML[14] is frequently used due to the ease of set-up.

When setting up a ML, the user, who will act as the ML manager, sets the mail address for the ML, and then registers the members. In general, participation and withdrawal of members are managed by the ML manager, and participation or withdrawal is done by sending a request to the manager. This endows authority to the manager, and while it enables removal of troublemakers, it also places a heavy responsibility on the manager.

Therefore, QuickML[15] was employed as our ML. QuickML is a ML system that enables set-up and operation with simple maneuvers.

4.2.2 QuickML

QuickML is a ML management system developed by Satoru Takabayashi and Toshiyuki Masui. It enables easy set-up and operation. In QuickML, all members have authority to determine participation/withdrawal of members, so the problem of placing a burden on a single manager does not occur. Mail sent in by a member is forwarded to all members, so each member will manage the mail as a client.

5 Implementation of qwikWeb

5.1 Implementation method

For implementation of qwikWeb, QuickML and Wiki were used as component. Implementation was done so the users would not be conscious of QuickML or conventional Wiki, and the system looked like an ingenious combination of the functions of ML and Wiki to users. In qwikWeb, knowledge structuring was promoted by registering all mails sent to ML as Wiki pages.

User can set up ML and Wiki site by simply sending e-mail to the qwikWeb system. First, the user selects a name for ML, and sends e-mail to "name-of-ML@qwik.jp." If the name is not already used, confirmation mail will be returned. By responding to this mail, ML and Wiki site is set up (Figure 2). If such ML and Wiki sites remain unused for one month or more, they are automatically deleted after issuing warnings. Almost no burden is placed on the user for starting and closing a site, and the user can start-up ML and Wiki by simply throwing in an e-mail.

The e-mail sent to the ML is automatically stored in the Wiki site as a new Wiki page. The title of the mail becomes the page name of the Wiki page, and in the case where the same page name already exists, the new mail is added to the end of the already existing Wiki page. Response to that mail is also added on to the end of the Wiki page of the parent mail. The mails within the same Wiki page are displayed divided by date and user name. Related mails are stored on one Wiki

Fig. 2 Newly set up Wiki.
Exchange on ML directly becomes Wiki page.

page. The Wiki site can be accessed only by members of the ML. Addition of members can be done by mail or Wiki. Addition of members can also be done by sending mail to the ML with cc mail addresses of members to be added. From Wiki, members can be added by directly editing the the member list of the Wiki page. Any member can take part in this member management maneuver, so no single manager is burdened.

In qwikWeb, Wiki plays the role of archiver, but unlike ordinary ML archiver, it can be edited because it is Wiki. For example, minor typing error that does not have to be notified by mail can be easily corrected by directly editing the Wiki page. Also, when mail "We want to set a date, so please check the day which is convenient for you" is sent to the ML, one can check the convenient day directly onto the mail on the Wiki page, rather than responding to the mail. When a file is attached to the e-mail, that file is shared on the Wiki.

With qwikWeb, exchanges on the ML can progress seamlessly to collaborative editing. Unlike e-mails, however, since not all members check the Wiki, some users may not notice that editing has taken place. Therefore, qwikWeb has a function where the system automatically and regularly sends update information of a Wiki page (of the name of user that did the update and the update time) to the ML. All Wiki pages are version controlled, so items can be recovered even if someone deletes them accidentally. Also, there is time machine function that allows the editing status of the Wiki page to be viewed easily. This is a function where one can see the sequential change over time of a Wiki page by moving the slide bar to left or right.

In user participation, there is always a concern for security. Therefore this system was designed with utmost care. For example, a mail address is used as ID and a password is automatically generated by the system. The password cannot be obtained by typing in someone else's mail address, so impersonation is difficult. In case one forgets the password, it is sent when one enters the mail address, but it is sent to the mail address entered so only the owner of the address may receive it. Of course, if the password is stolen or accidentally sent to someone else, impersonation becomes possible as in other systems. However, if the impersonator alters the website, the alterations are notified to the members, so the actual person will know someone else has engaged in unauthorized maneuver. Since all web pages keep past history, the undesired change can be restored to a desired state, and future impersonation can be prevented by deleting the ID registration. Recovery is also easy in case wrong entries are made since past entry history is stored.

5.2 Points of implementation

qwikWeb was developed, not just as a mere practical web system, but as "the ideal solution for a web system" and we tried to get close to the ideal as much as possible. Specifically, we followed the REST architectural style by Roy Fielding. In the beginning of qwikWeb development, there was no web application framework that could be used easily, so we created parts of the framework including templates and plug-ins. To keep installation and maintenance simple, we selected mechanism of using a file system only as storage rather than database.

In general, increasing stability as a system for practical service and adding functions occasionally as a system for research are two contradicting activities. If functions are added, bugs occur and the system becomes unstable. However, both must be done simultaneously to satisfy both research and practical aspects. To accomplish the two contradicting properties, we attempted to decrease bugs as much as possible by thorough function test descriptions. Since we created the web application framework ourselves this time, and also created the test framework, it was easier than usual to describe the function tests for the web system. Since a function test was described beforehand at all times during the development, there was almost no function without a function test. Hence, we were able to operate the system with hardly any length of downtime. For both development and operation, the Author (Eto) alone has been able to continue the operation for four and half years.

6 Operation of qwikWeb

6.1 Access analysis of qwikWeb

Using the implemented qwikWeb, research operation started in August 2003, and the system continues to be in operation as of present. Users must agree to the terms and conditions (such as data will be used for research purposes without identification of individual users) when using the system.

Figure 3 shows the cumulative values of the number of MLs, the number of closed MLs, and the number of MLs using Wiki. As of May 2007, there were 3,110 MLs and 18,519 users (excluding data for ML set up by developers). Maximum number of users in a single ML was 648 people,

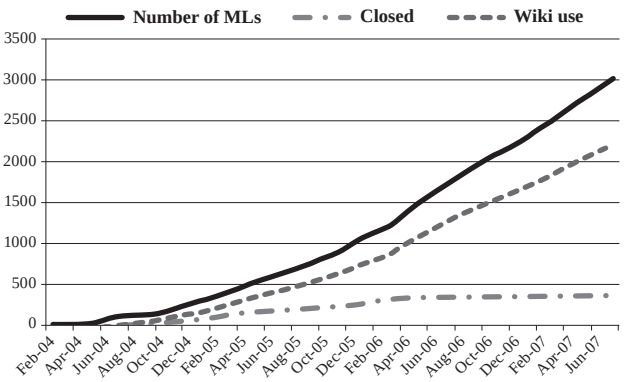


Fig. 3 Time transition of number of MLs.

and the average was 7.7 people. The majority were MLs with 10 or less people, and this matched the design intent of a ML to be used by a small number of people. Wiki was used in about 70 % of the MLs, and many users selected qwikWeb since Wiki can be also used. While the average active period of MLs using Wiki (2,235 MLs) was 145.1 days, it was 42.4 days for MLs without Wiki (875 MLs). It can be seen that Wiki was employed for long-term use. The number of closed MLs is currently about 10 %, and this is low compared to 50 % closed ML for QuickML, and a longer use is expected to promote more accumulation of knowledge.

The cumulative values of users are shown in Figure 4. Number of users, number of Wiki users, and number of users who created MLs all increased. About 70 % were MLs with Wiki, and about 40 % were users of Wiki. This is probably because many users started using Wiki by induction through the ML. About 10 % of users created MLs (1,909 people). Maximum number of MLs created by one person was 40 MLs, and the average was 1.6. As seen from the fact that there was a single user who created 40 MLs alone, the cost to the manager is relatively low. About 25 % or 505 people created 2 or more MLs. On the other hand, looking at the number of MLs in which the users participated, of the 18,519 people, the maximum number of participation was in 48 MLs, and the average was 1.3. About 20 % or 3,515 people belonged to 2 or more MLs. Participating in several MLs may be due to that fact that the users considered qwikWeb to be useful to some degree.

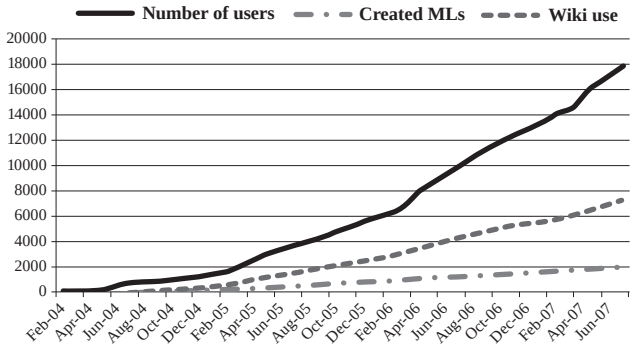


Fig. 4 Time transition of number of users.

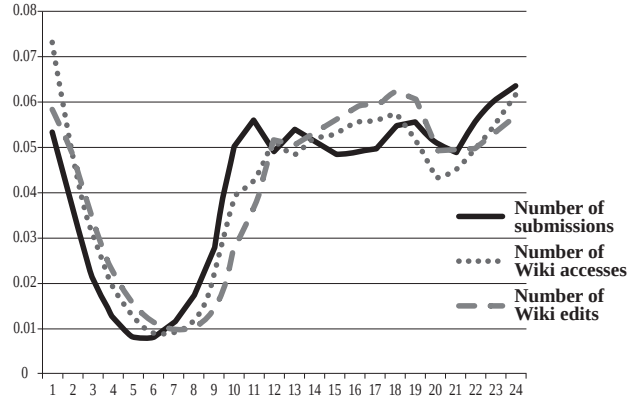


Fig. 5 Difference in use status of ML and Wiki by time zone.
The solid line of the graph shows the percentage of submissions by time, when the number of submissions throughout the day is set as 1.0. The same is applied to numbers of Wiki access and Wiki edit.

The difference between Wiki and ML was studied from the difference in use by time zone. Figure 5 shows the number of submissions, number of Wiki accesses, and number of edits to Wiki by time. It can be seen that peaks of submission to a ML occurred during 9:00~11:00, 12:00~13:00, 17:00~19:00, and 23:00~24:00. The time zones corresponded to start of work, after lunch break, end of work, and before retiring for the evening. In contrast, the number of Wiki accesses and edits increased from start of work in the afternoon to end of work, and reached a peak at 17:00~18:00. The reason for this is, since submission to a ML is communication, it was done most often at start and end of segments in daily routine, while access to and editing Wiki are part work so it was done during work in the afternoon, and as summary at end of work and before retiring for the evening.

Next, we looked at the use by groups that use both Wiki and ML. Here, we focused on MLs with over 50 submissions, 50 times or more editing of Wiki page, and durations of 1 year or more. Of the total 3,110 MLs, 64 MLs met this condition. Comparing the number of submissions and number of edits by month, about 80 % or 54 MLs had more months with higher number of submissions than edits. Figure 6 shows the cumulative values of number of submissions and edits in some MLs. Figure 6(a) is mainly ML and Figure 6(b) is mainly Wiki. Overall, the first few months were most active (with many submissions and edits) and the activities tended to decrease afterwards. Particularly Wiki showed that trend, and it tended to subside after initial heavy editing. In some MLs, the number of edits to Wiki suddenly increased. Figure 6(c1) and 6(c2) show steadily growing number of submissions, while the number of Wiki edits increased suddenly at some point and then subsided. It is thought that both ML and Wiki are regularly used if they are used for communication in daily work and for collaborative knowledge structuring, but when they are used for a particular event only, use increased suddenly before and after the event.

The use of qwikWeb and the difference in how ML and Wiki were used was discussed based on operation data. Although this was a research operation, the efficacy of qwikWeb design and implementation was demonstrated.

6.2 Example of qwikWeb in practice

Articles on Wiki appeared over 12 issues of Software Design of Gijutsu Hyoronsha, from August 2005 to July 2006. One article of the series was “Complete Guide to qwikWeb”[16]. At the same time, qwikWeb was used as a collaboration system to create the article series, and the actual process of

editing using qwikWeb was presented in the final article[17]. The qwikWeb group used in the article series was opened to public to serve as reference for similar users[18]. According to the article, the reasons qwikWeb was selected were: "overview of the project can be readily seen by members who joined later" and "both mail and document can be viewed." Here, the objective of qwikWeb was realized. From this example, it can be confirmed that qwikWeb reached a quality that could be used at a practical level even though this was a research run.

7 Related researches

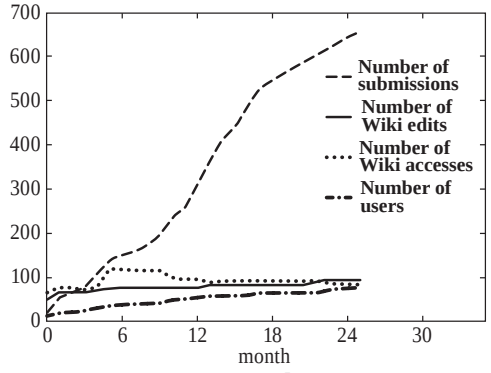
The basic concept of Wiki is that anyone can easily edit the contents, and there are several similar systems called Wiki clones that have adopted this concept. There are Wiki engines equipped with a function of linkage with mail to enable more convenient use of Wiki. In JotSpot[19], a Wiki page can be updated by sending e-mail to the mail address corresponding to each Wiki page. PukiWiki[20] allows an addition of a function of handling the mails received as entry to the page. Hiki[21] has a function of notifying updates by mail. All these use e-mail as a supplementary function of Wiki, but qwikWeb is different in the point that it is joined closely to ML. There are also Wikis with specialized purposes. Perhaps the most famous one is MediaWiki. This is a Wiki engine used in Wikipedia, the online encyclopedia. There are also several proposals for a Wiki engine called Semantic Wiki, a tool to easily create semantic web data[22][23]. There are Wiki engines for educational purpose. Guzdial et al developed a Wiki engine called CoWeb which is used in education[24]. In 2 years of operation, over 120 Wikis were started up in schools and are operated on 10 servers. Wang et al added extensions needed for learning environment such as page ownership, writing authorization, and invisible mode[25]. Brereton et al constructed an educational support environment based on Wiki[26]. However, there is no proposal of a system fused seamlessly with ML as in qwikWeb.

8 Summary

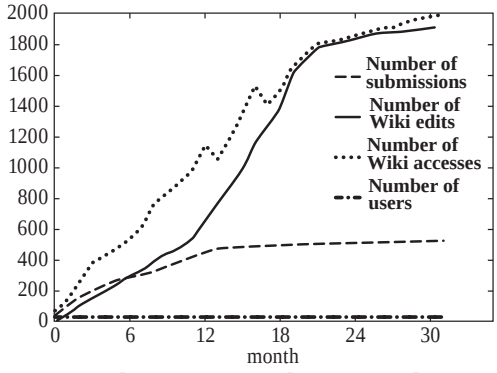
We proposed qwikWeb that smoothly fused information communication, accumulation of knowledge, and structuring technology to realize a collaboration system that can be built and managed by anyone. We described the process for realizing Full Research of such a collaboration system. As a result of designing, implementing, and operating this system, about 18,000 users used about 3,000 MLs. Adequacy and efficacy of this system were demonstrated by analyzing the operation data.

Future issues include commercial operation by companies, improvement by focusing on target users, and improvement for use in diverse information environment such as cell phones.

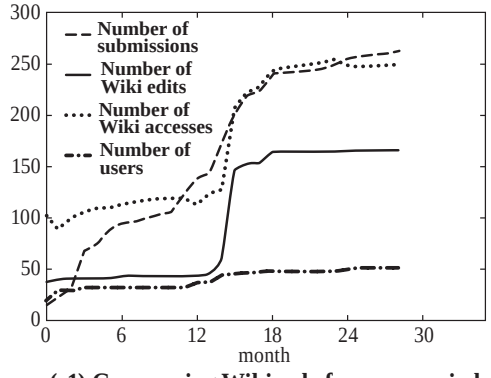
Modulobe, or virtual organism construction environment,



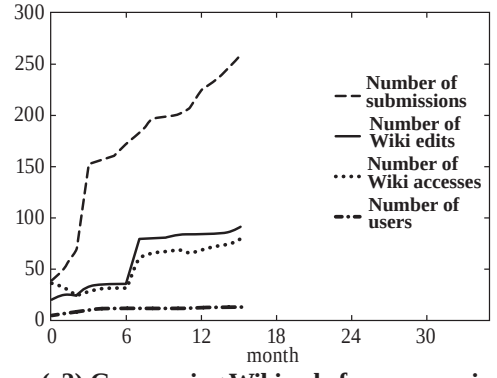
(a) Group mainly using ML



(b) Group mainly using Wiki



(c1) Group using Wiki only for some period



(c2) Group using Wiki only for some period

Fig. 6 Difference in use status by group.

is another project of the Author (Eto). In Modulobe, virtual organism created by the user is shared on the net, and other users can create new models by adding changes to the model. ID attached to the parts of the model allows the relationship of re-use of models to become visible. It is similar to qwikWeb since it is an attempt to seek findings by running a system that supports communication among users.

Both qwikWeb and Modulobe have common points of "design by subtraction." The most essential function for the user is considered, while extraneous functions are deleted completely. Therefore, the target users can understand the system more readily. In qwikWeb, comprehensibility was achieved by deleting functions that normally exist such as functions to assign managers or to change editing authority for each page. Although this may make qwikWeb unusable in some situations, we were able to create a system that new users can understand and use immediately.

In the future, we shall continue research that supports communication among users and contributes to knowledge sharing on the network.

Acknowledgment

Part of this research was supported by Research Topic "Construction of Citizen's Art Creation Platform by Information Design" in Research Field "Core Technology to Support Production of Digital Media Contents" in Core Research for Evolutionary Science and Technology (CREST) of the Japan Science and Technology Corporation (JST).

References

[1] Ministry of Internal Affairs and Communications: Results of 2005 "Survey on Use of Communications," http://www.soumu.go.jp/s-news/2006/060519_1.html (2006) (in Japanese).
[2] K. Eto, S. Takabayashi, and T. Masui: qwikWeb: a communication system combining mailing list and Wiki, *Interaction 2005*, 13-20 (2005) (in Japanese).
[3] Lotus Notes: http://www-06.ibm.com/jp/software/lotus/
[4] Cybozu: http://cybozu.co.jp/
[5] K. Okada: Communication support in collaborative work, *Journal of The Institute of Electronics, Information, and Communication Engineers*, 89(3), 213-217(2006) (in Japanese).
[6] S. Kitayama: Measuring the community, *Journal of The Japanese Society for Artificial Intelligence*, 18(6), 668-674 (2003) (in Japanese).
[7] B. Leuf and W. Cunningham: *The Wiki Way: Quick Collaboration on the Web*, Addison-Wesley, Reading, MA (2001).
[8] K. Eto: Why is Wiki so important? *Mobile Society Review Mirai Shinri*, 7, 50-57 (2006) (in Japanese).
[9] C. Alexander, S. Ishikawa and M. Silverstein: A Pattern Language: Towns, Buildings, Construction, Oxford University Press, New York (1977).
[10] K. Beck and W. Cunningham: Using pattern languages for object-oriented programs, Technical Report No. CR-87-43, http://c2.com/doc/oopsla87.html (1987).
[11] E. Gamma, R. Helm, R. Johnson and J. Vlissides: Design patterns: Abstraction and reuse of object-oriented design, *Proceedings of ECOOP'93*, Kaiserslautern, Germany, 406-431 (1993).
[12] http://c2.com/cgi/wiki?WikiWikiHyperCard
[13] Yahoo! Group: http://groups.yahoo.co.jp/
[14] freeML: http://www.freeml.com/
[15] S. Takabayashi and T. Masui: QuickML: convenient group communication tool, *IPSJ Journal*, 44(11), 2608-2616 (2003) (in Japanese).
[16] K. Eto: Complete explanation of qwikWeb, *Software Design May 2006 Issue*, 102-111 (2006) (in Japanese).
[17] S. Shibamura: Taking a bite out of Wiki on qwikWeb – Behind the scenes of Wiki Tsumamigui operation, *Software Design July 2006 Issue*, 113-115 (2006) (in Japanese).
[18] http://qwik.jp/wikibana-gihyo/
[19] JotSpot: http://www.jot.com/
[20] PukiWiki: http://pukiwiki.org/
[21] Hiki: http://www.namaraii.com/hiki/
[22] H. Takeda and M. Hendry: For the construction of semantic MediaWiki, *Study Group Reference Material of The Japanese Society for Artificial Intelligence*, SIG-SWO-A404-06, 06-01--06-03 (2004) (in Japanese).
[23] K. Kawamoto and Y. Kitamura: Automatic RDF generation by semantic Wiki, *Study Group Reference Material of The Japanese Society for Artificial Intelligence*, SIG-SWO-A501-02, 02-01--02-06 (2005) (in Japanese).
[24] M. Guzdial, J. Rick and B. Kerimbaev: Recognizing and supporting roles in CSCW, *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, 261-268 (2000).
[25] C. Wang and D. Turner: Extending the Wiki paradigm for use in the classroom, *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2, 255-261 (2004).
[26] M. Brereton, J. Donovan and S. Viller: Talking about watching: Using the video card game and wiki-web technology to engage IT students in developing observational skills, *Proceedings of the Fifth Australasian Conference on Computing Education*, 197-205 (2003).

Received original manuscript December 26, 2007
Revisions received March 10, 2008
Accepted March 10, 2008

Authors

Kouichirou Eto
Graduated from Graduate School of Media and Governance, Keio University in 1997. Joined the International Media Research Foundation in 1997. Produced work as media artist for themes of communication on the network. Produced "WebHopper" in 1996. Won the Grand Prix of Ars Electronica Award in 1997 (as sensorium project). Produced "Sound Creatures" jointly with Canon ARTLAB in 1998. Received Honorary Mention of Ars Electronica Award in 1999. Joined the AIST in 2002. Started development and operation of qwikWeb in 2003. Launched virtual creature construction environment Modulobe in 2005. Interested in information sharing environment to support knowledge sharing on the web. Master of Science (Media and Governance). Member of the Information Processing Society of Japan. In this paper, mainly worked on design,

development, trial run of qwikWeb, and writing.

Masahiro Hamasaki
Graduated from Faculty of Engineering, Doshisha University in 2000. Completed master's program at Graduate School of Information Science, Nara Institute of Science and Technology in 2002. Completed doctorate program at School of Mathematical and Physical Science, Graduate University for Advanced Studies in 2005. Doctor of Science (Information Science). Joined the Information Technology Research Institute, AIST in 2005. Works on information commendation and online community. Interested in information system utilizing human network. Member of the Information Processing Society of Japan (IPS); The Institute of Electronics, Information and Communication Engineers (IEICE); and The Association for Computing Machinery (ACM). In this paper, worked mainly on data analysis of qwikWeb trial run and writing.

Takuichi Nishimura
Completed master's degree (Measurement Engineering) at Faculty of Engineering, The University of Tokyo in 1992. Joined NKK in 1992. Worked on R&D for x-ray and sound/vibration control. Dispatched to the Real World Computing Project (RWCP) Tsukuba Research Center in 1995. Returned to NKK in 1998. Joined the RWCP Tsukuba Research Center in 1999. Joined the Cyber Assist Research Center, AIST in 2001. Became leader of Real-World Interaction Group, Information Technology Research Institute, AIST in 2005. Also works as assistant professor of Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba. Doctor of Engineering (Engineering). Interested in time-course data search/identification and real world information support. Member of the Information Processing Society of Japan (IPS); The Institute of Electronics, Information and Communication Engineers (IEICE); The Japanese Society for Artificial Intelligence; Human Interface Society; The Japan Society of Infrared Science and Technology; and The Association for Computing Machinery (ACM). In this paper, mainly worked on positioning of the research, theoretical development, and writing.

Discussion with reviewers

1 On improvements for user participation

Question (Kazuhito Ohmaki)

I think the perspective that "it is very important to have a mechanism where end users can become directly involved in system design" is important, and I think this point should be clearly stated.

Answer (Kouichirou Eto)

As you indicated, since the improvement process with involvement of users is important in information systems, I explained further in "2. Full Research for web system." Although direct involvement of end users is very important, in this development, we employed a method where the users use the system, we, the developers, observed them carefully, and then we thought of the necessary functions based on our observations. Since users are novice of the system, they cannot necessarily propose truly necessary functions. However, since end users are specialists in their own respective work, the developers determined the necessary functions based on their work processes. This point was described in Chapter 2.

2 On rights of software

Question (Kazuhito Ohmaki)

Have you clarified the rights for qwikWeb?

Answer (Kouichirou Eto)

The people who were involved in direct development of qwikWeb were AIST researchers only, and the relationship of rights is clear. Although QuickML had been used as part of the system, it is disclosed as free software based on General Public License (GPL). qwikWeb also is under GPL, there is no rights issue.

3 On learning how to use softwares

Question (Kazuhito Ohmaki)

Having used Wiki, the Reviewer feels that although simple, one must learn "pattern language" to some point. I would like to hear your comments on what you think of the trade-off.

Answer (Kouichirou Eto)

Thank you for pointing this out. First, although pattern language is background philosophy of Wiki, it does not mean that one cannot use Wiki without learning it. Anyone can start using it by learning the simple maneuvers. To make full use of Wiki, one must learn pattern language to some extent, as you point out. That means that one must learn the background philosophy of Wiki, but one can learn it gradually. In other words, by using Wiki, one can gradually become acquainted with the philosophy of pattern language, and that is one advantage of Wiki.

In Wiki, it is necessary to express the sentence structure by using Wiki syntax, which is its unique mark-up language, and the time required to learn this could not be ignored. However, by writing sentences using Wiki syntax, one becomes conscious of the sentence structure. In fact, when I teach Wiki in college courses, by teaching Wiki as well as advantage of structuring sentences at the same time, the students were able to learn Wiki syntax naturally. Such trade-off does exist for Wiki syntax.

As you pointed out, concerning this point, I added explanations.

4 On goal of development, scenario, and selection process of component

Question (Motoyuki Akamatsu)

In the last part of Section 2.3, you mention four points. Explanations assumed the fusion of ML and Wiki, except in the first point. I think it will be better to write a scenario by describing the four points as four goals of system design, and then to write that the Authors decided to fuse ML and Wiki after considering how to achieve the goals (Description of Scenario). Also, there must have been a selection process of component, and I think you should write the "selection process" where certain technologies other than ML and Wiki were not employed.

Answer (Kouichirou Eto)

The scenario, as you point out, was not clear in the original paper, so the chapters were arranged according to the process in Figure 1, and changed to reflect the flow of development from problems and solution strategy of current collaboration ("ease of management" and "accumulation and structuring of knowledge"), basic philosophy toward solution, selection of Wiki that had concept close to our basic philosophy, and employment of QuickML in basic design of the system. For selection processes, the reason for selecting Wiki as component is explained in 4.1.1, Wiki is explained in 4.1.2, the reason for selecting Quick ML as component is explained in 4.21, and Quick ML is explained in 4.2.2.

5 On process of improving operation

Question (Motoyuki Akamatsu)

In "4. Operation of qwikWeb," you present the result of user data analysis. I think this section corresponds to the cycle of analysis of feedback information, system correction, and system operation, but there is no description of how the correction to the system was done according to this cycle. Can you clarify so the relationship with the improvements can be seen?

Answer (Kouichirou Eto)

I explained an example by adding "6.2 Practical example of qwikWeb." The feedback here corresponds to the explanation in Figure 1.

6 On difficulty of research and development

Question (Motoyuki Akamatsu)

Reading this paper, I get the impression that qwikWeb was created easily. I imagine that there were various problems, and it could only be created after solving those problems. I think it will benefit the readers greatly if you include descriptions of technological difficulties and how they were solved.

Answer (Kouichirou Eto)

I described the problems we had in implementation and operation in Section 5.2.