Title	A knowledge management system for dynamic organizational knowledge circulation
Author(s)	Hirai, Chiaki; Uchida, Yoshinobu; Fujinami, Tsutomu
Citation	International Journal of Information Technology and Decision Making, 6(3): 509-522
Issue Date	2007-09
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/7921
Rights	Electronic version of an article published as International Journal of Information Technology and Decision Making, 6(3), 2007, 509-522. DOI:10.1142/S0219622007002617. Copyright World Scientific Publishing Company, http://dx.doi.org/10.1142/S0219622007002617
Description	

A knowledge management system
for dynamic organizational knowledge circulation

Chiaki Hirai and Yoshinobu Uchida
Systems Development Laboratory, Hitachi, Ltd.
Tsutomu Fujinami
Japan Advanced Institute of Science and Technology
E-mail: {c-hirai, yosinobu}@sdl.hitachi.co.jp
fuji@jaist.ac.jp

Abstract

The demand for knowledge management is increasing because knowledge is an important and essential resource for sustaining competitiveness. We report a system to support knowledge asset reuse. Incorporating work breakdown structure-based project management, workflow management, and a document database, the system guides workers to efficiently store and reuse knowledge assets. To dynamically circulate knowledge throughout an organization, the system supports the concept of knowledge flow, through which knowledge is transferred from one project to another. A distinctive feature of our approach is that after collecting relevant knowledge, the system sends it to the members of a project and prompts them to have a meeting to discuss whether they accept the knowledge. The purpose of this meeting is to have members internalize knowledge assets and share their tacit knowledge through discussion. We developed a prototype system and evaluated it using our experience with it.

Key words: knowledge management, work breakdown structure, workflow, knowledge flow, document management, process knowledge

A knowledge management system for dynamic organizational knowledge circulation

Abstract

The demand for knowledge management is increasing because knowledge is an important and essential resource for sustaining competitiveness. We report a system to support knowledge asset reuse. Incorporating work breakdown structure-based project management, workflow management, and a document database, the system guides workers to efficiently store and reuse knowledge assets. To dynamically circulate knowledge throughout an organization, the system supports the concept of knowledge flow, through which knowledge is transferred from one project to another. A distinctive feature of our approach is that after collecting relevant knowledge, the system sends it to the members of a project and prompts them to have a meeting to discuss whether they accept the knowledge. The purpose of this meeting is to have members internalize knowledge assets and share their tacit knowledge through discussion. We developed a prototype system and evaluated it using our experience with it.

Key words: knowledge management, work breakdown structure, workflow, knowledge flow, document management, process knowledge

1. Introduction

The demand for managing knowledge is increasing in knowledge-intensive organizations. This is because knowledge has come to be considered an important and essential resource for sustaining the competitiveness of an organization. At the same time the definition of knowledge remains an unsolved issue, and the concept of knowledge management covers a wide area from relatively simple IT-based data sharing systems to knowledge creating process models such as the SECI model [NONAKA].

Our approach is to build an IT-based knowledge sharing system that enables storing and reusing knowledge in an organization. The hypothesis that knowledge is of two types, explicit and tacit [NONAKA], is now widely accepted. Though an IT system cannot handle tacit knowledge directly, tacit knowledge is the key to knowledge innovation [KROPH].

Our primary target is to support R&D laboratories since they are typical knowledge-intensive organization and have definite needs for knowledge management. We assume that an R&D laboratory has several projects running concurrently and that knowledge should be shared among those projects. There are two types of project management methods; one of which is work breakdown structure (WBS) based management and the other of which is workflow based management. The former defines the static work structure and executes a project as planned, while the latter manages a project dynamically depending on the situation according to predefined flexible work procedures. We developed a system that enables knowledge asset reuse, incorporating WBS-based project management, workflow management, and a document database. Our system guides workers to efficiently store and reuse knowledge assets. In order to dynamically circulate knowledge throughout an organization, the system also implements the concept of knowledge flow [NISSEN], through which knowledge is transferred from one project to another. A distinctive feature of our approach is that after collecting relevant knowledge, the system sends it to the members of a project and prompts them to have a meeting to discuss whether they accept the knowledge. The purpose of this meeting is to have members internalize knowledge assets and share their tacit knowledge through face-to-face discussion.

Hence, the key function of our system is navigation. It directs workers to do their jobs in optimized orders. They are given the proper information to execute a job, encouraged to register documents produced in the job, and at the proper time, they are prompted to share their tacit knowledge through discussion.

We have so far developed a prototype system based on the above concept. In Section 2, we examine issues and problems of conventional knowledge management systems. After reporting on the system we developed in Section 3, we evaluate the system in Section 4.

2. Integrated knowledge management system

2.1. Process-based knowledge management

The early stages of IT-based knowledge management produced only document sharing systems. However, the focus of this field is shifting to process-based knowledge management. Documents are surely essential knowledge resources in an organization, but organizational knowledge exists in the process as well as documents. This is the knowledge of how to work, or you could say, the algorithm describing how to behave in a certain situation. Thus, the process dominates the behavior and eventually dominates the performance of an organization. This is because the process has recently been accepted as an important knowledge asset. A process-based knowledge management approach typically consists of process definition, process execution to guide workers according to a process, and continuative process improvement. One of the proposals reported in this paper is regarding process definition and process navigation.

2.2. Explicit knowledge and tacit knowledge

There are two types of knowledge: explicit and tacit. Explicit knowledge can be expressed in language and is relatively easy to transfer from person to person. This knowledge can be stored in a computer system as written documents or data. Tacit knowledge, on the other hand, cannot be expressed in language. Knowledge of this type is transferred only by direct human contact, typically through face-to-face discussions.

As described in the previous section, we believe that processes are organizational knowledge. We call this "process knowledge" in this paper. Process knowledge is "how-to" knowledge, which describes procedures or methods to achieve tasks. On the other hand, what is produced as result of a process and used as input information for other processes is called "artifact knowledge.".

From this point of view, we have to deal with four types of knowledge: explicit artifact, tacit artifact, explicit process, and tacit process. Our research targets both types of explicit knowledge and tacit artifact knowledge. Tacit process knowledge remains for future work.

2.3. Our basic approach to knowledge management

Our approach is to build an IT-based knowledge sharing system to enable storing and reusing knowledge assets in an organization. The key function of our system is navigation. The system guides workers to efficiently store and reuse knowledge assets. Our sharing model, however, is not a stock model, which has places for users to save knowledge and take it from. Our model is a project-based knowledge flow model. The basic idea is that knowledge properly flows from project to project according to its content and the characteristics of the projects. Our system supports this concept. The system has a process, or a set of ordered jobs, for each project and guides workers to execute jobs in that order. To help execute a job, the system collects all relevant documents produced in other projects and provides the documents to the worker responsible for the job. The documents made during the job are stored as the result of the job and are transferred to other projects that need those documents. Thus, the system controls the flow of knowledge among projects.

A distinctive feature of our approach is that after collecting relevant knowledge, the system sends it to the members of a project and prompts them to have a meeting to discuss whether they accept the knowledge. The purpose of this meeting is to have members internalize knowledge assets and share their tacit knowledge through face-to-face discussion.

According to the SECI model, explicit knowledge is internalized by individuals before they share tacit knowledge. However, the problem is that if a whole organization begins to store its explicit knowledge in a document database, the amount of knowledge will soon exceed the limit for internalization. An individual or a team has to select relevant knowledge from the knowledge database, which is an important issue in knowledge management and is called knowledge selection [HOLSAPPLE]. This can be half-done by using information technology, such as text mining, a concept search, or automatic document categorization. However, our hypothesis is that if a team screens and selects useful documents for the team through discussion, this discussion would be a good opportunity to internalize knowledge assets and share tacit knowledge. Because selecting documents requires a basis of values, teams have to clarify and share their basis. We call the meeting where this is done a “knowledge screening meeting.” Our system directs workers to have knowledge screening meetings, which is how we deal with tacit knowledge. We cannot handle it directly, but we can manage meetings to share tacit knowledge.

In summary, we propose an IT system that has the following characteristics:

- It stores explicit process knowledge, or work processes, and guides workers according to the processes.
- It provides workers with relevant documents as input for the work and stores documents produced during the work. These documents are the explicit artifact knowledge in our system.

- Each work process requires a screening meeting, and the system guides members to have those meetings, which help members share tacit artifact knowledge.

2.4. Problems of conventional process-based management systems and our approach

Systems already exist to manage work processes, and some trials have been performed on integrating work process management and document management [AVERSANO]. Those conventional process management systems are basically of two types. One is work breakdown structure (WBS) [HUMPHREY] based project management systems, and the other is workflow systems. In this section, we describe the essential characteristics and problems of both systems and propose a way to integrate WBS and workflow from the knowledge management point of view.

The problem of WBS is in its maintenance during a project. WBS-based project management requires a precisely defined WBS prior to the project. However, actual projects rarely develop as planned. Consequently, the WBS often has to be updated. This necessity to maintain the WBS is troublesome for project leaders and is an obstacle to introducing this kind of management system. This problem is caused by the simplicity of the WBS model, which is defined in a tree structure and has no control mechanisms such as selection or loops. Hence, WBS cannot tolerate any deviations from its precise definition without maintenance.

Workflow, on the other hand, defines the work procedure with a process definition language, which is essentially a programming language with a control mechanism. The work structure can vary depending on the status of work if the process has been programmed appropriately.

We combined the WBS-based approach and the process language approach to define work processes and to guide workers because we believe that a WBS and process language are completely different in their purposes and that they complement one another.

First, we need to precisely examine the definition of process. Process algebra defines a process as a set of all possible event sequences. A specific event sequence that actually happened is called a trace of a process [HOARE]. Though workflow is a process in this sense, a WBS is not a process in the process algebra sense. A WBS is an expected trace of a process, namely, a planned trace. The reason process management tools often adopt a WBS is that a project needs a plan, or an ideal event trace, and a WBS is a way to define an event trace, not a process.

An organization often defines a standard WBS and includes it in the knowledge assets of the organization. However, as we examined above, a standard WBS is not a standard process, but a standard plan of work. A standard plan that is too detailed is not practical. There's a dilemma here. A detailed standard plan would guide workers well, but, at the same time, it is vulnerable against environmental changes. The conventional artifice to overcome this problem is to add ad-hoc processes requiring human decisions. For instance, the standard process defines a mandate procedure A, but also defines the situation that exceptionally allows procedure B. This artifice, however, is error-prone, if the situation and ad-hoc rules get complex.

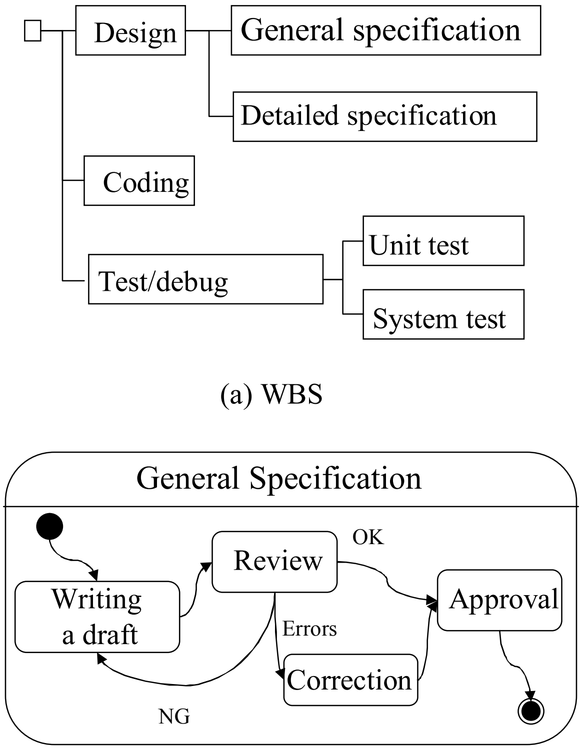
We propose that only the overall schedule be defined as a WBS and that a standard WBS should be a standard overall schedule. On the other hand, a conventional workflow system does not include the concept of an overall schedule. It can define a very precise and detailed process, but the general flow of the whole project is hard to comprehend from a detailed workflow. Thus, describing a whole project with workflows is not practical.

Based upon the examination above, our approach to defining a work procedure is a combination of a WBS and workflow. A WBS triggers work components one by one according to its work structure, and each work component is executed according to its workflow. This method balances the rigidity of the general schedule and the flexibility of the small granularity workflow. If sufficient types of work components are pre-defined as the organization's standard work components, a leader can plan the project's WBS just by selecting from the work components.

3. Our knowledge management system

3.1. WBS, workflow, and document management integration

First, we systematized all the documents in our organization. We collected all types of documents and defined unique type identifications (type-IDs) for each document type. This is the key information for integrating the system. As described in the previous section, a WBS defines the general schedule of a project and is a collection of work components, of which the detailed procedure is defined as a workflow. Second, we analyzed all types of work components in our organization and defined each work component using a formal model. We adopted the state transition model according to Tsuji's examination [TSUJI] and extended the idea with the object-oriented methodology. A state, or a node, of the state transition model, is a work package that has the type-IDs of the documents to be made in this work package and references, which are URLs in our system, to be referred to complete the work package. A work package in our system is a *class* in the sense of the object-oriented methodology and can generate instances inheriting the class's attributes. When a state transition occurs, the state, or the work package, generates an instance of the work package. The instance has the URLs that were defined in the class work-package as references and has instances of documents to be made that inherit type-IDs from the class work package. An example of a WBS and a work component defined by a workflow are shown in Fig. 1.



(a) WBS

(b) Work component

Figure 1 Example of WBS and work component.

Carrying out a throughout survey on our organization's work variations, we stored all the types of work components obtained by the survey in a work component database. This database enables a project leader to build his/her project's WBS just by selecting needed work components from the database and defining the proper order. We also defined some organizational standard WBSs for some typical types of projects. Thus, we have two levels of explicit process knowledge: the standard WBS is knowledge of how to run a project, and the work component is knowledge of how to execute a small granularity job. The first unique feature of our approach is the idea of how to combine these types of knowledge systemically, as shown below in Fig. 2, which shows the overall architecture of our system.

At the beginning of a project, the leader constructs the project's WBS by either using a predefined standard WBS or selecting work components. The leader then inputs the name of the person responsible for each work package, the starting date, and the deadline for the work package. As the project develops, the WBS triggers work components one by one according to the defined order, and a work component generates work packages according to the state transition model. Generated work packages are stored in an issue management system until they are completed. This issue management system sends reminder e-mails to the person responsible for a work package, namely, the author of the document to be made in the work package. The timing and content of the reminder e-mail are defined as reminder rules with respect to the type-ID of the document. The issue management system also provides a status report that shows the list of completed works and ongoing works.

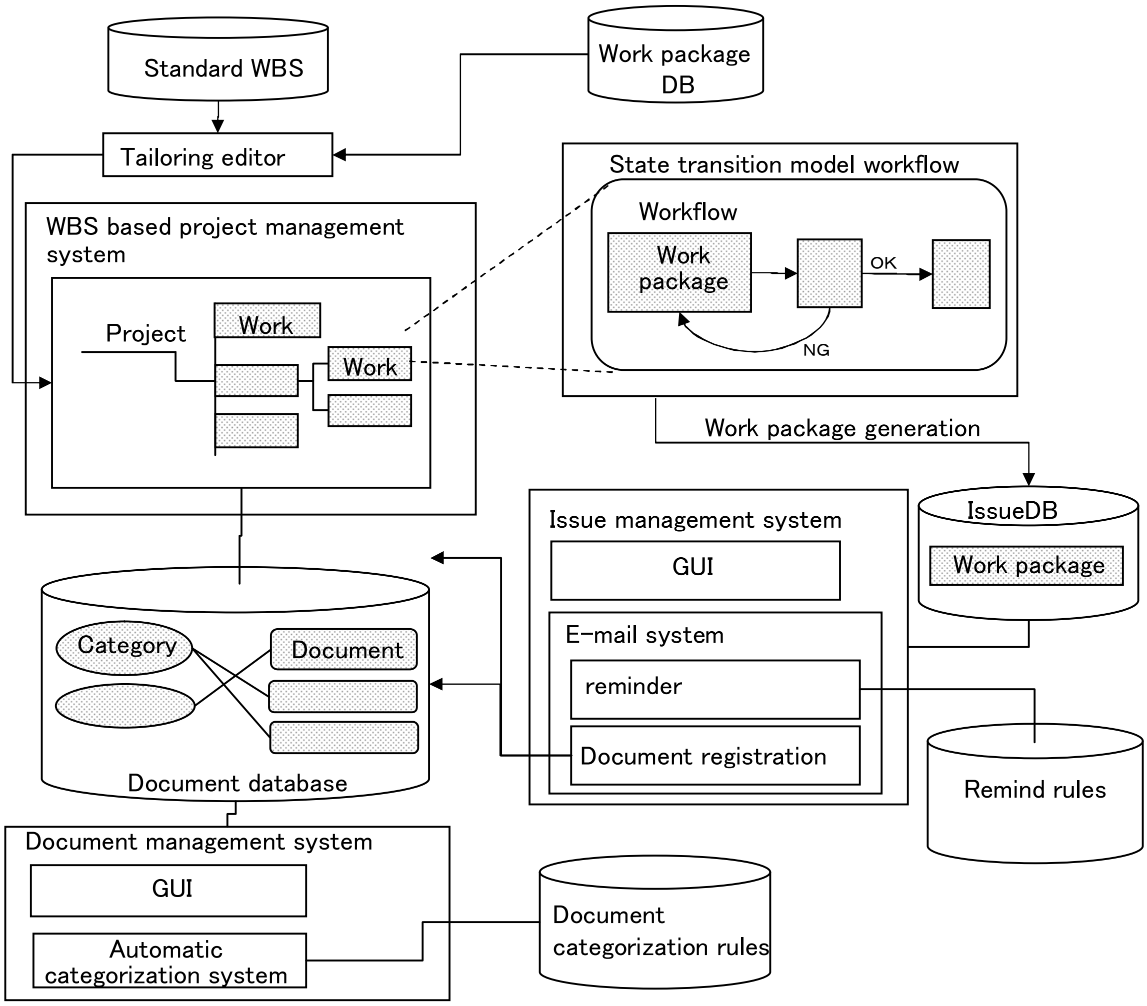


Figure 2 System architecture.

When the author finishes the document, he/she replies to the e-mail and attaches the document. The system retrieves the document from the e-mail and stores it in a database. Because the system knows which reminder e-mail the reply corresponds to from the e-mails' reply-to header, the system can specify which work-package instance and document instance correspond to the stored document, and thus can specify the *origin* of the document: the project, the work component, and the work package that the reply belongs to and the type-ID of the document.

From project members' points of view, they are guided by e-mails to execute their jobs in a proper order. Reminder e-mails are sent repeatedly until the document is submitted. The fact that members can submit documents by e-mail deserves attention. They don't have to learn a new document management system to register documents. This removes the first and often the biggest obstacle to introducing a document sharing system into an organization. Thus, explicit knowledge can be shared more easily.

3.2. Knowledge flow and screening meetings

The second feature is the idea of knowledge flow and screening meetings described in Section 2. A work component is not independent from other work components. For example, if a WBS has a work component to write a specification document and a work component for programming, the output of the former will be the input to the latter. This is an inner knowledge flow, which can be implemented using references in our system. The system adds the specified document's URL to a work package in the programming work component as a reference.

Another example is a work component to write a survey report on available technologies and a work component to write a patent on a new technology. A knowledge flow from survey work to patent writing is reasonable. However, in this case, a knowledge flow across the projects would be suitable because the survey report, made as part of another project, may be useful when writing the patent. Hence, the connection between the survey work and the patent work is not bound to a project. The implementation of this cross-project knowledge flow is again done using references. Collecting all survey reports written in the organization, the system sends them to the patent writing work.

In our system, we define two types of knowledge flow between work components: the project knowledge flow that transfers documents from one work component to another in a project and the organizational knowledge flow by which documents are transferred across projects.

Examples of knowledge flow are shown in Fig. 3. Boxes represent work components, and arrows represent both types of knowledge flows. For example, the patent work component has a knowledge flow to itself. This means that the component obtains all of the patent documents across the organization as references, which means that the person who writes the patent is expected to check all of the previously written patents in the organization.

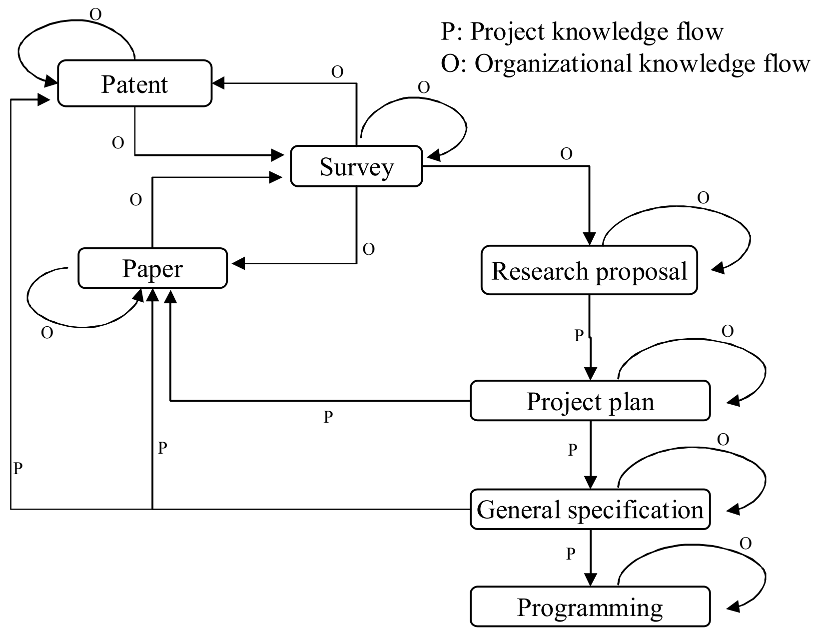


Figure 3 Examples of knowledge flow.

The problem here is that too many documents are input to a work component. In the patent case above, all of the survey reports in the organization are collected, but only some reports would be relevant for writing the specific patent. Though some conventional WBS-based project management systems support project knowledge flows by defining dependency relationships between jobs, organizational knowledge flow has not been proposed because of this problem. As a solution, we introduced a screening meeting as the first work package of every work component (Fig. 4). The purpose of this meeting is to select documents relevant to the project's focus. The person responsible for this meeting calls necessary project members, and they discuss which documents are relevant to the project.

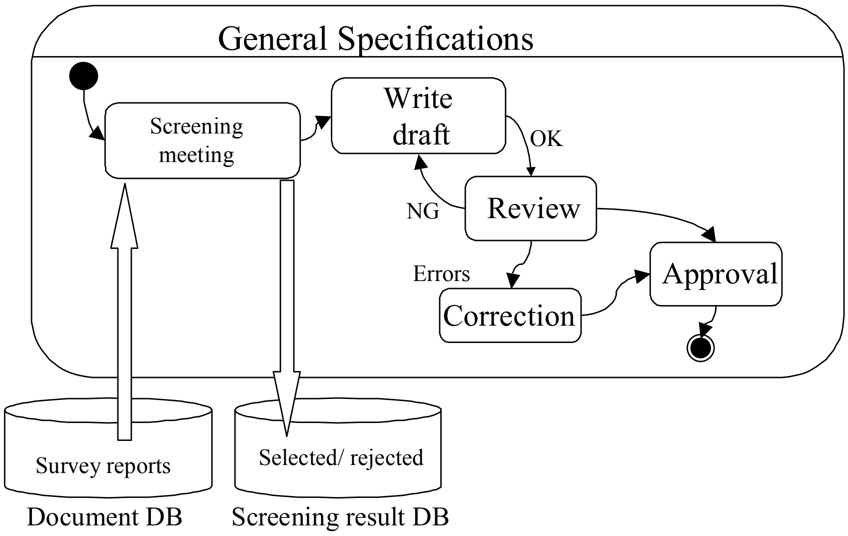


Figure 4 Screening meeting.

Selecting documents through discussion is important. Through discussion, members can share their tacit background knowledge related to the documents. Moreover, they have to clarify and share their basis of values for selecting relevant documents. This basis depends on the purpose of the project, which means that they have to clarify the purpose or the nature of their project through discussion. Hence, we believe that this discussion works as an opportunity for the participants to internalize organizational knowledge assets, share their tacit knowledge, and reiterate the purpose of the project.

The system adds the URLs of the collected documents to an instance of the screening meeting work package as references and sends an e-mail to prompt the person responsible for the screening meeting to hold it. The output of the screening meeting is a list of selected documents and a list of rejected documents, both of which are submitted to the system via e-mail.

The system stores both lists, which are going to appear as references in every work package in this project Succeeding work components in the project can pre-screen the collected documents using these two lists. For example, in Fig. 5, Project A has a screening meeting in a research proposal work component to select relevant survey reports from the document database. They will have a screening meeting for survey reports again during the survey work. However, the second time, they will not need to discuss the survey reports again. The system will tell them which were selected and which were rejected. They will only need to discuss documents that were written after their first screening meeting.

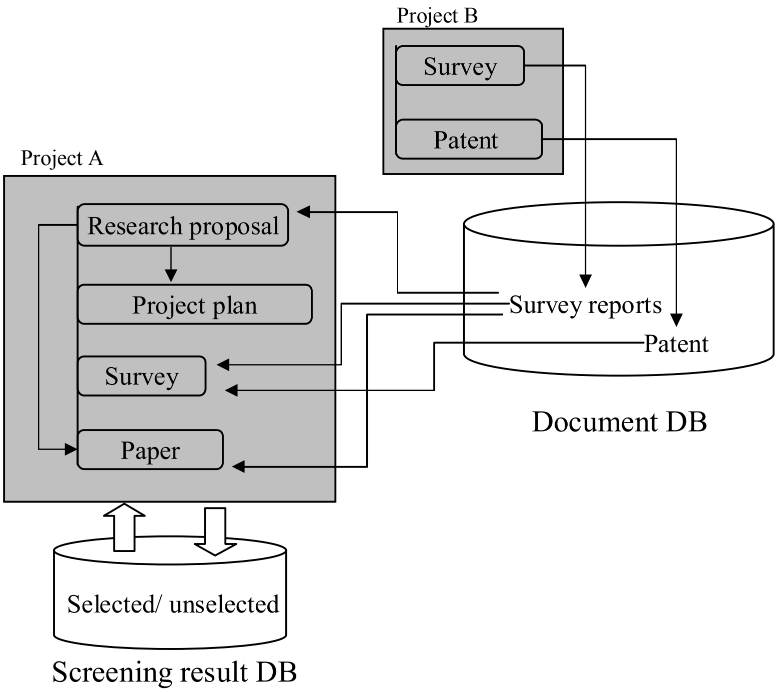


Figure 5 Knowledge flow between projects.

4. Evaluation of the system

We developed a system and have been running it for six years. In this section, we describe what we observed during our experience and evaluate the effectiveness of the system.

1) Documents have actually been shared.

We believe this is one of our most important achievements. Sharing documents in an organization is not easy. Workers are reluctant to register documents because they have to input additional information such as the types of documents or which project the documents belong to. Without this information, a document database cannot categorize documents and hence cannot present useful views. Under our system, a WBS-based project management system, a workflow system and a document management system were integrated on a common document type-ID system. The system knows the origin of each document when a project leader constructs the project WBS. The message IDs of e-mails exchanged between the system and the author is the link information to connect a document and its origin. Our approach enables workers to submit documents just by replying to e-mails, thus removing the main obstacle to document sharing. Fig. 6 shows the number of documents stored per year in a single model team. The sixth year only stored 1097 documents by 18 members, which were almost all documents written in this model team. he access to this document base was 1000-1500 every month.

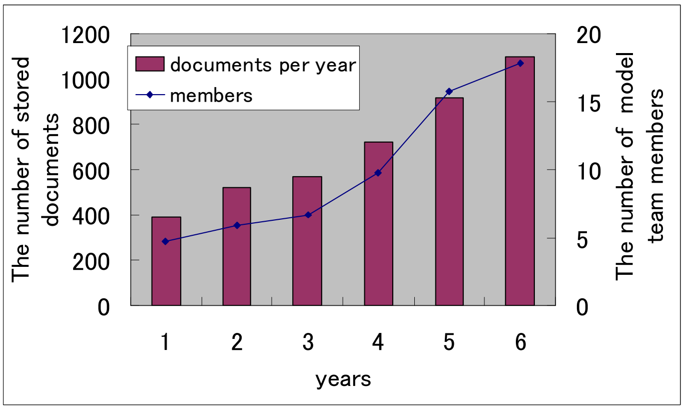


Figure 6. The number of documents stored per year.

2) The process improvement cycle was shortened.

Standard WBSs could be often revised just by exchanging some components. And since the process improvement involves both the changes of work procedures and document-format changes, the integration of workflow management and document management made it easier to redesign a standard process. As a consequence, this enabled a project leader to define a trial new process and see whether it works well, which helped the organization find better work process quickly.

3) We developed a method to define knowledge flow across an organization.

Often an IT-based knowledge sharing approach neglects the importance of face-to-face meetings. An IT system can have a large number of documents, but people are often reluctant to use it because the database is not guaranteed to have the information needed and no perfect search engines exist to enable users to search for what they really want. In our approach, the system prompts members to have a meeting to discuss the content of the database. Though we have not yet checked the effectiveness of our method, participants of screening meetings had positive impressions in spite of the fact that the meetings often took all day. Screening information by oneself would be a boring job, but discussion was not and was sometimes exciting. Participants surely shared their knowledge. The results of the discussions, namely, the lists of selected documents, were circulated in the projects members. This circulation in most cases prompted some feedback from members who did not attend the meeting. Thus, the lists worked to extract related knowledge and increased the overall amount of knowledge shared.

4) Knowledge flow is a new feature added to the knowledge map.

The knowledge map, which shows which parts of an organization have what kind of knowledge, is a useful tool to get an overall view of the knowledge assets of the organization [EPPLER]. Our system brings a new feature to the knowledge map. At a screening meeting, documents relevant to a project are selected. This selection decides the valid knowledge flow from one project to another. If a project selects many documents produced in some other project, these two projects can be considered to have a knowledge flow between them. This flow map is useful, for example, to evaluate who is creating valuable knowledge assets and how the assets are used in an organization. The concept of knowledge flow between projects is shown in Fig. 7. This is the dynamic aspect of the knowledge map.

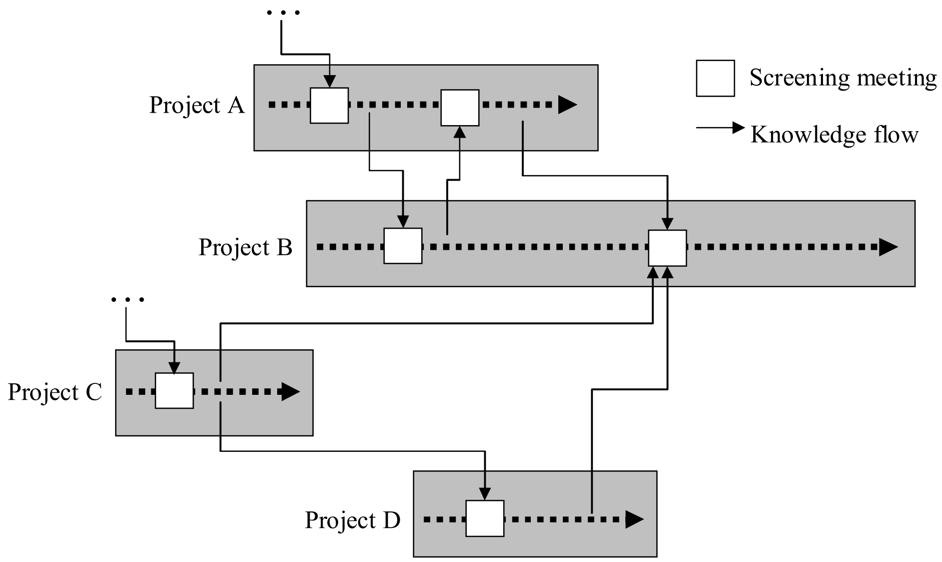


Figure 7 Knowledge flow concept.

5) The improving potentials of this system

Our proposal to realize the concept of knowledge flow is based on the assumption that the screening meeting is essential to absorb knowledge from outside. But through the usage of this system, we found that it depends on the situation whether or not the meeting should be held. At present, project leaders judge the situation.

But there may be some criteria or guidelines for the judgments. Organizational literature indicates that "boundary spanner" is the key role in the interaction between organizations. The basic idea is that an organization communicates with much different-natured organizations through some limited number of individuals, called boundary spanners [ALLEN 1969, 1977; TUSHMAN]. According to this theory, an R&D team should have a gatekeeper to obtain information from, for example, a market division [TUSHMAN], but when communicating with other R&D teams, the screening meeting could be a better option.

This issue needs further research, which, we expect, will lead to new functions on our system.

5. Conclusions

We developed a knowledge management system. Our approach is to integrate a WBS-based project management system, workflow system, and document database into a single knowledge management system. The bond that connects those component systems is the type-ID of documents.

The leader of a project can build the general schedule, or the WBS, simply by selecting work components, which are defined using the state transition model. This state transition model guides project members by sending e-mails indicating the next step. Members can register the results of their work simply by replying to the e-mails. Thus, documents can easily be accumulated in the document database. We also introduced the knowledge flow concept into our system. Before a work component is executed, all the relevant documents are collected across an organization and are input to the workflow. The system prompts project members to have a meeting to select from the collected documents those they need for their project. Through this discussion, members can share their tacit knowledge. The result of the discussion is reused to assure the efficiency of the meetings in the project. We have developed a system based upon the above concept and have been using it. Our experience shows documents are effectively shared using our method?]. Though the effectiveness of the knowledge flow has yet to be proved, it has been used to visualize the dynamic aspect of knowledge transfer, and thus, it is a new feature on the knowledge map.

Reference

[ALLEN 1977] Allen, T. J.: Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization. MIT press, 1977.

[ALLEN 1969] Allen, T. J. et al.: Information Flow in Research and Development Laboratories, Administrative Science Quarterly, Vol.14, pp.12-19, 1969.

[AVERSANO] Aversano, L. et al.: Integrating Document and Workflow Management Tools using XML and Web Technologies: a Case Study, Proceedings of the Sixth European Conference on Software Maintenance and Reengineering (CSMR'02), 2002.

[EPPLER] Eppler, M. J. et al.: Making Knowledge Visible through Knowledge Maps: Concepts, Elements, Cases, Handbook on Knowledge Management 1, pp. 189–205, Springer, 2003.

[HOARE] Hoare, C. A. R.: Communicating Sequential Process, Prentice-Hall, 1985.

[HOLSAPPLE] Holsapple, C. W. et al.: Knowledge Selection: Concepts, Issues, and Technologies, Knowledge Management Handbook, pp. 7-1–7-17, CRC Press, 1999.

[HUMPHREY] Humphrey, W. S.: Managing the Software Process, Addison-Wesley, 1989.

[KROPH] Kroph, G. et al.: Enabling Knowledge Creation—How to Unlock the Mystery of Tacit Knowledge and Release the Power of Innovation, Oxford University Press, 2000.

[MALAFSKY] Malafsky, G. P.: Technology for Acquiring and Sharing Knowledge Assets, Handbook on Knowledge Management 2, pp. 86–107, Springer, 2003.

[NISSEN] Nissen, M. et al.: Dynamic Models of Knowledge-Flow Dynamics, CIFE Working Paper No.76, Stanford University, 2002.

[NONAKA] Nonaka, I. et al.: "The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation," Oxford University Press, Oxford, 1995.

[TUSHMAN] Tushman, M. L.: Managing Communication Network in R&D Laboratories, Readings in the Management of Innovation, Ballinger, pp.261-274, 1988.

[TSUJI] Tsuji, H. et al.: ELISE: Office Procedure Automation Tool by State-Transition Model,

Journal of Information Processing, Vol. 12, No. 1, pp. 9–16, 1989.