Designing Knowledge Management Systems: Reuse and Integration of Findings in Computer Supported Cooperative Work

Kibum Kim, Deborah G. Tatar
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA
kikim@vt.edu, tatar@vt.edu

Abstract

Currently, we lack the technical mechanisms to fully bridge the gulf between the social requirements and technological feasibility. Many interactive systems today are developed without the involvement, input, or advice of trained psychologists or human factors experts, who might best be able to forecast the way users will respond to new or different designs. In this paper, we analyze the prospects for knowledge management (KM) systems with respect to the social and technical issues discussed and illustrated in Ackerman's nine general findings of computer-supported cooperative work (CSCW). Based upon this examination, we then develop several hypothetical KM scenarios to illustrate the CSCW concerns. Finally, we discuss the nature of the social-technical gap for KM systems and suggest what designers can do to reduce it.

1. Introduction

In an effort to manage knowledge effectively, business organizations worldwide continually search for and implement a variety of techniques and technologies. One such tool, the knowledge management system, can be defined as a systemic and organizationally-specified process for acquiring, organizing, and communicating an organization's implicit and explicit knowledge to enhance effectiveness and productivity [2]. This ability to integrate and apply specialized knowledge proves crucial to a firm's ability to create and sustain a competitive advantage. As a result, more companies are becoming acutely aware of how important it is not merely to "know what they know" but also to be capable of maximizing such knowledge by focusing on its codification, collection, integration, and dissemination.

However, because of the natural gap between what we need to do socially and what we can do technologically (i.e., the social-technical gap), many problems remain between identifying knowledge and being able to use and manage it in an efficient, cost-effective manner. Human activity is highly flexible, nuanced, and contextualized; therefore, computational entities such as information transfer, roles, and policies must be similarly flexible, nuanced, and contextualized. Some of the requirements that a knowledge-management support system should address include [1]:

- Having an unified vocabulary that will ensure shared knowledge is correctly understood;
- Being able to identify, model, and explicitly represent this knowledge;
- Sharing and reusing knowledge among different applications for various types of users (This notion implies being able to share existing knowledge sources, as well as future ones.); and
- Creating a culture that encourages knowledge sharing.

In this paper, we analyze the prospects for knowledge management systems with respect to the social-technical gap discussed and illustrated in Ackerman's paper [3]. Section 2 briefly summarizes state-of-the-art KM system technologies, with a selection of three examples, while Section 3 discusses the implications of Ackerman's nine general findings of computer-supported cooperative work on knowledge-management concerns and activities. Based upon this examination, we then develop several hypothetical KM scenarios to illustrate the CSCW concerns. Finally, based on Sections 2 and 3, Section 4 discusses the nature of the social-technical gap for KM systems and suggests what designers can do to reduce it.

2. State-of-the-art technologies in current KM systems

KM systems possess an extensive history. To facilitate the coding and transmitting of knowledge, industries have long relied on a number of traditional forms, such as training and employee development programs, organizational policies, routines, procedures, reports, and manuals. However, in order to systematize, facilitate, and

0-7803-8819-4/04/$20.00 ©2004 IEEE.

expedite firm-wide knowledge management, modern KM systems can take advantage of altogether new and exciting methodology: embedded IT assembly, including Internet, intranets, browsers, data warehouses, data filters, data aggregation, direct manipulated graphical user interface (GUI) and software agents. To identify state-of-the-art technology in current KM development and to suggest how it might be used to map a design space for future systems, this section examines three related systems.

The first system is WISEngine's IDB, a virtual database platform that assembles in real time data and applications from multiple Internet or intranet data sources and provides a standard query interface across multiple disparate data sources [4]. The second one—RETH software, by Siemens Inc.—combines goals and functional requirements in a scenario-based design process. We investigate RETH could be used effectively for knowledge management; in fact, investigating such potentiality benefits the future design goals of KM systems [5]. Finally, this section will study the ASSISTUM Knowledge Tool, created by Assistum Ltd., which offers a suite of knowledge management tools that rely on artificial intelligence and fuzzy logic to deepen the user's understanding of commercial and practical issues affecting an important decision [6].

2.1. IDB system

The key component of any KM system is an integrated and integrative technology architecture. To function effectively, KM systems require a variety of technological tools in three areas: database and database management, communication and messaging, and browsing and retrieval. The need for seamlessly integrating the various tools in these areas could lead to the dominance of Internet and intranet-based KM system architectures [2].

The WISEngine's IDB system enables a uniform view of multiple heterogeneous data sources, such as HTML/XML-based Internet data and intranet data stored in RDBs, mail servers, and file systems, among others. It enables real-time data aggregation, transformation, and queries based on standard SQL. One of the most exciting IDB technologies involves a novel approach called "soft-wrapping," which wraps information sources. Because conventional wrapping methods require an extensive programming process to define an application interface, it can take 2-5 days for an experienced programmer to wrap a single application. Conversely, the IDB system's unique SDL (Source Description Language)—which creates shorter scripts, generally 5-10 lines in length, using a GUI tool—dramatically increases the efficiency of the wrapping process. A novice end-user programmer can wrap one application in a mere 7-10 minutes. In addition, instead of being pre-compiled into the system or "hard-wrapped," source descriptions can be executed at run time.

Soft-wrapping possesses several advantages [7]. First, as source descriptions are written independently of the run-time environment, it proves more flexible and portable than hard-wrapping. To provide additional portability, the IDB system is written in Java. Second, without having to restart the system, soft-wrappers can be tested and registered dynamically at run-time through a Web interface. Third, because recompilation is not needed, soft-wrapping adapts easily to dynamically-changing Web data sources. Finally, soft wrapping is more secure: rather than being a pre-compiled wrapper that must be trusted by whoever executes it, what is registered is instead a declarative query.

The IDB system illustrates one possible approach to how end-user programming (e.g., soft-wrapping) can be embedded and used in a KM system design space. Figure 1 shows the IDB system architecture.

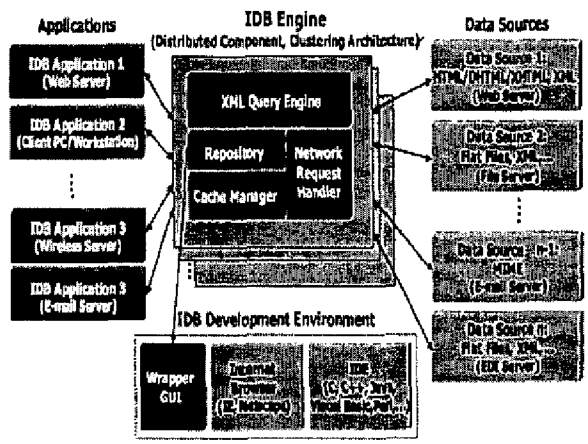

Figure 1. The IDB system architecture

2.2 RETH (Requirements Engineering Through Hypertext)

In general practice, organizations must discover leverage points where enhanced "knowledge" can add value, and then develop a KM system that will deliver the required knowledge [2]. Developed by Siemens Inc., RETH is both a method for requirement engineering and a tool supporting this method [5]. Based on hypermedia (hypertext and multimedia integrated) and object-oriented technology, RETH uses several technologies to deliver required knowledge: hypertext links, automatic object-associations, and combination models. First, RETH generates hypertext links between requirements, scenarios and goals; that is, it automatically creates and determines the source and destination of requirement chains. Second, in order to support the creation of an initial object-oriented model, RETH also automatically generates associations between objects. Finally, and most importantly, RETH combines scenarios with both functions and goals. With its emphasis on scenarios, this combination model examines how effectively the functions of a designed system will serve the goals of its users. By relating the goals, scenarios, and functions of both the current and the new system, one can track the evolution of how people complete a task. RETH confirms the utility of scenarios and extends their use in the designing of a KM system. Figure 2 illustrates a typical RETH window.

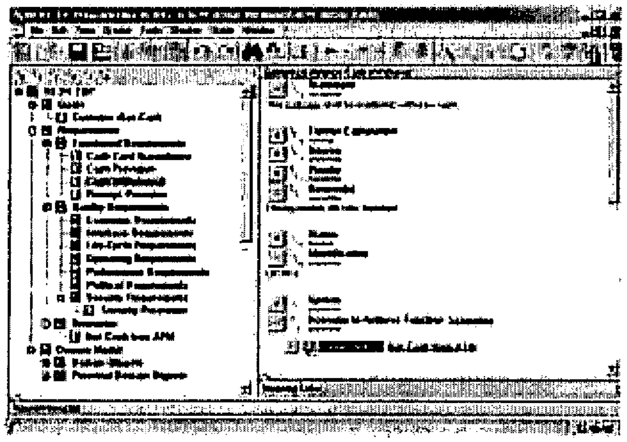

Figure 2. A typical RETH window for requirement analysis

2.3. ASSISTUM knowledge tool

ASSISTUM enables the efficient creation and modification of knowledge bases that can be easily accessed by organizations. More than that, using ASSISTUM, organizations can capture, enhance, and build their own organization knowledge assets [6]. The system's two main components, coded with Visual Basic, are the *Assistum Knowledge Viewer* and the *Assistum Knowledge Editor*. The former enables users to interact with knowledge bases in order to develop judgments and conclusions, while the latter permits them to create or modify knowledge bases. Figure 3 depicts a knowledge network delineating the potential for teaching. Green links support teaching; red links deny it. The thickness of the lines indicates the importance of the link's impact.

ASSISTUM possesses an artificial intelligence (AI) or "fuzzy logic" to store and transform a node's degree of truth (DoT), then to compute the impact on the DoT of any dependent nodes. Using fuzzy logic, the DoT is weighted by a range from 0 to 1 and calculated by membership functions. ASSISTUM indicates how effectively AI can be used in the KM process.

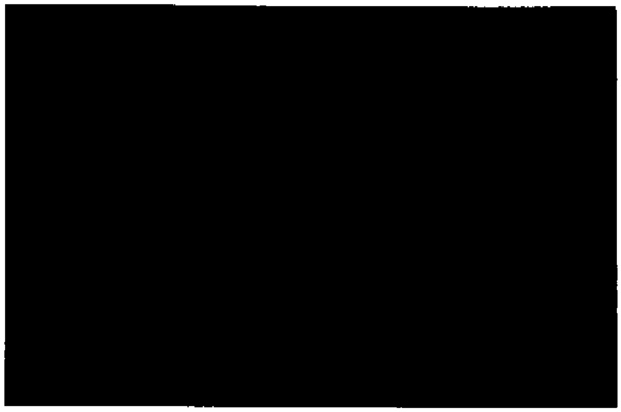

Figure 3. Knowledge network about the potential for teaching

Clearly, such state-of-the-art technologies as IDB, RETH, and ASSISTUM indicate the shape and direction of future knowledge management systems from a technological perspective. However, the design of a technical system has always effects on the human aspects and the other way around. Since such sociotechnical systems involve users, with their distinctive personality traits and social interaction, that future must also take into consideration the social factors introduced by Ackerman's nine general findings of computer-supported cooperative work. In other words, in addition to understanding a system's technological architecture, we must also learn how its social framework will affect its use.

3. Ackerman's CSCW findings

In his pioneering work, Ackerman (2001) summarizes nine general findings of computer-supported cooperative work [3]:

1. Social activity is fluid and nuanced, yet systems often have considerable difficulty handling such detail and flexibility.
2. If there are hidden or conflicting goals, people will resist concretely articulating their own expectations. On the other hand, with shared meanings or histories, people work effectively at resolving communicative and activity breakdowns. CSCW should find ways to manage the problems and tradeoffs resulting from conflict and coordination.
3. Exceptions are normal in work processes. CSCW should deal with exceptions and fluidity.
4. People use awareness to guide their work. CSCW should consider tradeoffs inherent in issues of awareness-versus-privacy and awareness-versus-disturbing others.
5. Visibility of communication exchange makes learning more efficient, but it may also make work

more formal and reduce sharing. CSCW should determine ways to manage the tradeoffs in sharing.

6. Users often actively negotiate the norms for using a CSCW system. Such systems should have some secondary mechanisms or back-channel communication methods to support negotiation.
7. With an insufficient number of users, people will not use a CSCW system. The critical mass problem for CSCW systems should be solved.
8. People not only adapt to their systems, but they adapt their systems to their needs (co-evolution). CSCW system designers should assume co-evolution will occur.
9. Incentives are critical. CSCW researchers should try to provide a suitable organizational reward structure.

In this section, we develop several succinct hypothetical scenarios of KM to illustrate these CSCW concerns. In addition, drawing on various Human-Computer Interaction (HCI) models and frameworks, we perform a claims analysis for each scenario, identifying a range of possible positive and negative consequences.

3.1. Scenario for Ackerman's finding 1

The KM system for Macrosoft company allows only two basic roles for the system's developers: *usability engineers*, who design general frameworks, and *software programmers*, who do real implementation. However, in reality, most usability engineers have some programming skills, and software programmers also possess a general concept of design.

- Allowing two discrete role components as entities of KM system :

  + enables one to view components easily; the purpose of entities is readily inferred. (Notational Systems [8])

  - BUT there may be a gap between what usability engineers provide (general frameworks) versus what software programmers want (guidelines for specific decisions). (Theory-Practice Gap [9])

  - BUT social nuances and flexibility in categorizing people may not be handled appropriately. (Social-Technical Gap [3])

3.2. Scenario for Ackerman's finding 2

This afternoon, Bill wants to retrieve information about a submarine volcano from the on-line teachers' community. He joins a science discussion group and checks the logged history of previous messages to catch up with the discussion.

- Providing the history of discussion to be checked by new participants :

  + helps to build common knowledge among dialogue partners and new participants. (Common Ground [10])

  + makes people good at resolving communicative and activity breakdowns. (Situated Action [11])

  - BUT there may exist social pressure on speakers, because all participants' comments are recorded and reviewed later. (Conformity Pressures [12])

3.3. Scenario for Ackerman's finding 3 and 6

John permits only his colleagues who teach science to access to the science class directories in his KM system. However, one day Susan, who teaches French, tries to access the file in his directory for her class. Because she is denied access, she e-mails John. To allow her access, John sets up his directory configuration again.

- Allowing exceptions in document sharing policy:

  + enables one to change plans according to moment-to-moment interpersonal coordination. (Situated Action [13])

  - BUT inappropriate or incorrect configuration changes may enable access to John's other directories also. (Hidden Dependencies [8])

3.4. Scenario for Ackerman's finding 4 and 5

For her colleagues, Nancy wants to post previous exams on the sharing documents repository in a KM system. She does not want to her name to be visible to others. However, for security reasons, the KM system requires that all posters identify themselves.

- Providing identity of each poster instead of anonymity:

  + prevents the occurrence of social loafing, which comes from anonymity. (Social Psychology [12])

  + boosts the individual's participation because of the uniqueness that derives from individual contributions. (Social Psychology [12])

  - BUT it may decrease the individual's willingness to contribute to sharing because of evaluation apprehension. (Social Psychology [12])

3.5. Scenario for Ackerman's finding 7

Three months ago, Jim set up a KM system for teachers in Blacksburg. He tried to make a whiteboard and a conferencing system active but for the first two months he needed to invite more users.

- Starting a KM service in the new organizations:

  - requires paying the start-up costs (gathering enough members) necessary to establish common ground. (Common Ground [10])

  + BUT to develop an information "scent," it encourages foraging. (Adaptation to the Environment [14])

3.6. Scenario for Ackerman's finding 8

For a long time Kim has used a Web customization system. When she got a new KM system, she tried to map her knowledge to it by relying on her previous experience with Web customization tools. After several trials, she finally found that the two interfaces differed markedly.

- Addressing the new KM system interface based on her extant skill and understanding:

  + rapidly engages knowledge and copes with novel situations. (Active User [15])

  + even in case of mismatches, evokes mental work; surprise, curiosity, challenge, elaboration, and insight. (Active User [15])

  - BUT may create bad inferences, overgeneralizations, and functional fixedness, thus blocking the imagination. (Active User [15])

3.7. Scenario for Ackerman's finding 9

In the document sharing system, Ann can find ample resources regarding diet foods, so she decides to subscribe and read the contributions. However, writing a contribution requires considerable effort, so she seldom produces one.

- Sharing documents equally among all users:

  + creates or enables mutual knowledge about shared information. (Common Ground [10])

  - BUT users can passively read without actively participating. (Free Rider [12])

4. Discussion

The above scenarios suggest that with regard to KM systems an obvious social-technical gap exists. Since it deals with how users manage their private knowledge in connection with other people, companies, and institutions, this gap between what is required *socially* and what we can do *technically* is inherently a CSCW and an HCI problem. Moreover, as previously noted, human activity is highly nuanced and contextualized, which presents additional problems [3].

For any agent developing a KM system, these issues create several concerns. For example, without a completely accurate grouping mechanism, in any situation few users can be categorized without errors (social ambiguity). Also, people in organizations and groups often possess conflicting goals, so any KM systems that ignore this aspect are likely to fail (social nuance). Additionally, if users who receive benefits from the KM systems are not the same as those who do the work, those systems will slide into disuse (social loafing) [12]. More generally, Grudin [16] mentions five social-technical gaps that lead to the failure of groupware applications:

- **Who benefits?**: Groupware applications often fail because they require some people to do additional work, and those people are not the ones who perceive a direct benefit from the application.
- **Compatibility with social practices:** Groupware may bring about activity that violates social taboos, threatens existing political structures, or otherwise demotivates the very users crucial to its success.
- **Exception handling:** Groupware may fail if it does not allow for the wide range of exception-handling and improvisation that characterizes much group activity.
- **Difficulty of evaluation:** We fail to learn from experience because these complex applications introduce almost insurmountable obstacles to meaningful, generalizable analysis and evaluation.
- **Poor intuitions for groupware:** The groupware development process fails because our intuitions are especially poor for multi-user applications.

Ackerman suggests that we should deal with this gap by ameliorating its effects and attempting to gain a better understanding of it. As a practical program of action, it is possible to devise several design issues for a KM system to reduce the gap [17]:

- KM systems should aim for some level of symmetry. For example, if one has to perform work for the system, one should obtain some benefit from it. This might involve improving the personal user interface, so that there are definite advantages in the quality of the work environment, user acceptance, and affect (e.g., the system is fun to use).

- To avoid the free rider problem in knowledge sharing system, we must increase the visibility of participants' contributions. For example, one could display an activity meter showing the number of contributions from each participant.
- To solve the critical mass problem, we must design KM systems so that they have benefits even when their user population is small.
- To better take into account hidden or conflicting goals between people, before developing a new KM system we must identify the stakeholders likely to be affected by it.

Currently, we lack the technical mechanisms to fully bridge the gulf between the social and technical worlds. Many interactive systems today are developed without the involvement, input, or advice of trained psychologists or human factors experts, who might best be able to predict the way users will respond to new or different designs. Thus, because computer system developers themselves cannot be expected to bring psychological expertise to the task, the basic psychology must be "built into" the models [18]. Although the social-technical gap is unlikely to go away, it certainly can be better understood and possibly dealt with. The challenge of the social-technical gap creates an opportunity to refocus knowledge management systems as creative and effective ways of supporting community.

5. References

[1] A. Macintosh, I. Filby, J. Kingston, and A. Tate, "Knowledge Asset Road Maps", In *Proceedings of The Second International Conference on Practical Aspects of Knowledge Management (PAKM98)*, Basel, Switzerland, Oct. 1998, pp. 29-30.

[2] M. Alavi, and D. Leidner, "Knowledge management systems: Issues, challenges and benefits", *Communications of the AIS, 1(2es)*, Article 7, 1999.

[3] M. S. Ackerman, "The intellectual challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility", *Human-Computer Interaction, 15(2-3)*, 2000, pp. 179-204.

[4] WISEngine Inc. Retrieved from the World Wide Web: http://www.wisengine.com/e_index.html, 2004.

[5] H. Kaindl, "Combining goals and functional requirements in a scenario-based design process", In *Proceedings of Human Computer Interaction'98 (HCI'98)*, Sheffield, UK, September, 1998, pp. 101-121.

[6] ASSISTUM Ltd. Retrieved from the World Wide Web: http://www.assistum.com, 2004.

[7] J. Kang, M.L. Lee, and J.F. Naughton, "IDB: Toward the scalable integration of queryable internet data sources", *Technical Report*, University of Wisconsin-Madison, 2000.

[8] A. Blackwell, and T. Green, "Notational systems-the Cognitive Dimensions of Notations framework", In J.M. Carroll (ed.) *HCI Models, Theories and Frameworks: Toward a multidisciplinary science*, Morgan Kaufmann, San Francisco, CA, 2003, pp. 103-134.

[9] Z. Pylyshyn, "Some remarks on the theory-practice gap", In J.M. Carroll (ed.), *Designing interaction. Psychology at the human-computer interface*, Cambridge University Press, Cambridge, MA, 1991, pp. 39-49.

[10] A. Monk, "Common ground in electronically mediated communication: Clark's theory of language use", In J.M. Carroll (ed.) *HCI Models, Theories and Frameworks: Toward a multidisciplinary science*, Morgan Kaufmann, San Francisco, CA, 2003, pp. 265-289.

[11] L.A. Suchman, *Plans and situated actions: The problems of Human-Computer Communication*, Cambridge University Press, Cambridge, MA, 1987.

[12] R.E. Kraut, "Applying social psychological theory to the problems of group work", In J.M. Carroll (ed.) *HCI Models, Theories and Frameworks: Toward a multidisciplinary science*, Morgan Kaufmann, San Francisco, 2003, pp. 325-356.

[13] B.A. Nardi, "Studying context: A comparison of Activity Theory, Situated Action Models, and Distributed Cognition", In B.A. Nardi (ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, MIT Press, Cambridge, MA, 1995, pp. 69-102.

[14] P. Pirolli, "Exploring and Finding Information", In J.M. Carroll (ed.) *HCI Models, Theories and Frameworks: Toward a multidisciplinary science*, Morgan Kaufmann, San Francisco, CA, 2003, pp. 157-192.

[15] J.M. Carroll and M.B. Rosson, "Paradox of the active user", In J.M. Carroll (ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, MIT Press, Cambridge, MA, 1987, pp. 80-111.

[16] J. Grudin, "Why groupware applications fail: Problems in design and evaluation", *Office: Technology and People, 4(3)*, 1989, pp. 245-264.

[17] A. Dix, J. Finlay, G. Abowd, and R. Beale, *Human-Computer Interaction*, Prentice Hall Europe, Hertfordshire, England, 1998.

[18] B.E. John, and D.E. Kieras, "Using GOMS for user interface design and evaluation: Which technique?", *ACM Transactions on Computer-Human Interaction, 3(4)*, 1996, pp. 320-351.