System Support for Knowledge Work:
Bridging the Knowing-Doing Gap

Dick Stenmark, Göteborg University, Sweden

Rikard Lindgren, Viktoria Institute, Sweden

ABSTRACT

Many researchers argue that information systems (IS) can play an important role in supporting organizational knowledge application. However, recent IS research indicates that knowledge management systems (KMS) often fail when implemented in the knowledge work practice of contemporary organizations. While KMS maintenance has been recognized as an important IS research area, imbalance between the desire for accurate content and the workload required to achieve this still appears to be a critical issue, resulting in systems of little use for organizations in their knowledge application processes. Driven by the ambition to contribute recommendations for how to integrate KMS with everyday knowledge work, we use general lessons learned from development of groupware applications as a theoretical lens to analyze empirical experiences from three implemented and evaluated KMS. Targeting the KMS maintenance challenge, our recommendations extend earlier IS research on the implementation and use of knowledge work support systems. On a practical level, our recommendations assist KMS developers in attempts to bridge the knowing-doing gap in organizations where individual members do not know or know of each other and the organization as a whole does not know what it knows.

Keywords: *group support systems; knowledge management systems*

INTRODUCTION

Over the last 10 years there has been much debate in academic literature about concepts such as knowledge-based organizations, knowledge-creating companies, knowledge work, and organizational knowledge (Blackler, 1995; Nonaka, 1994; Schultze, 2000; Spender, 1996). Consistent with this debate, knowledge management (KM) has been promoted as an important approach for organizations trying to achieve competitive advantage (Hedlund, 1994). Knowledge management is often regarded as the generation, representation, storage, transfer, transformation, application, embedding, and protecting of organizational knowledge (Schultze & Leidner, 2002). While

processes of knowledge generation, storage, and transfer do not necessarily result in improved organizational performance, effective knowledge application does (Alavi & Leidner, 2001).

According to the knowledge-based theory of the firm, the source of competitive advantage resides in the ability of an organization to turn knowledge into action and less on knowledge itself (Grant, 1996). Integration of knowledge, either explicitly or implicitly, of many different people to facilitate knowledge application, Grant argues, is the motivation for organizations comprised of multiple individuals. Recognizing that integration of knowledge of organizational members is exceptionally difficult, Grant advocates that a key challenge for organizations to achieve effective knowledge application is to establish a mode of interaction facilitating that people's specialist knowledge is integrated.

As noted by academics such as Davenport and Prusak (1998), there are several reasons for knowledge workers not to apply their knowledge. Chief amongst these are social factors such as distrusting the source of knowledge or lack of time or opportunity to apply knowledge (Alavi & Leidner, 2001). Observing that organizations tend to have a gap between what they know and what they do (Pfeffer & Sutton, 2000), many IS researchers suggest that information technology can have a positive influence on knowledge application (e.g., Alavi & Leidner, 2001). For example, information systems can enhance knowledge application by facilitating the capture, updating, and accessibility of organizational information and knowledge (Mao & Benbasat, 1998). Also, information systems can increase the size of knowledge workers' internal social networks by allowing for organizational knowledge to be applied across time and space (Kock & McQueen, 1998).

However, while contemporary organizations typically expect knowledge management systems (KMS) to become major innovations in terms of the ways in which business can be organized and be conducted, recent IS research indicates that such systems often fail when implemented in everyday knowledge work (Schultze & Boland, 2000). Despite the fact that KMS maintenance has been acknowledged as an important issue (Hahn & Subramani, 2000; Holtshouse, 1998), imbalance between the desire for accurate content and the workload required to achieve this still appears to be a critical problem, leading to systems of little use for organizations in their knowledge application processes (Lindgren & Stenmark, 2002). Following this, an important area of KMS research is the development of systems with the potential to bridge the knowledge application gap (Alavi & Leidner, 2001). In this context, a significant challenge is to develop design principles intended to keep KMS alive — updated, current, maintained — by encouraging use (Markus, Majchrzak, & Gasser, 2002).

The problems KMS are facing today, for example, the fact that systems remain unused in day-to-day practice despite good theoretical reasons why they should work, show great resemblance to the difficulties experienced when introducing groupware applications in the 1980s. Being one of the first to study the challenges faced by groupware developers, Grudin observed that when groupware started to emerge as a new market, many of the early application developers were people who previously had focused exclusively on single-user applications. The maturing single-user application domain forced these developers to explore new territories and pushed them into areas in which they had little knowledge. The problems they ran into they had never experienced when supporting individuals, and they were thus completely unprepared (Grudin, 1994). We believe that Grudin's observations are analogous to what we now witness in the knowledge management arena, where software vendors are being accused of re-labeling their old information systems to KMS (King, 1999), and that his influential work within the field of computer-supported cooperative work (CSCW) (Grudin, 1987; 1988; 1994) can prove helpful to KMS developers.

Despite these similarities and although KMS as organizational-wide technologies has

been discussed in terms of groupware (for example, Alavi and Leidner, 2001; Robertson, Sørensen, & Swan, 2001), Grudin's findings seem to be overlooked in the knowledge management literature. One of Grudin's chief findings is that situations where one party does the work and someone else receives the benefit often lead to failure. With activities on top of their ordinary responsibilities, organizational members cannot be expected to spend time and effort feeding a "knowledge database" or maintaining a "knowledge system" for the benefit of the organization only (Stein & Zwass, 1995). Recognizing that contributions from all organizational members are an important prerequisite for successful KMS (Hahn and Subramani, 2000), there must be mechanisms to express or represent knowledge in ways that also enable the individual employee to make better use of his or her knowledge (Kankanhalli, Tan, & Wei, 2005).

In this paper, we draw upon empirical experiences from three implemented and evaluated KMS at Volvo Information Technology AB in Sweden. For the purpose of contributing recommendations on how to integrate KMS with everyday knowledge work, we shall here use Grudin's (1994) eight challenges for groupware developers as a theoretical lens to analyze why the systems studied failed. Targeting the KMS maintenance challenge (Markus et al., 2002), our recommendations extend earlier research on the implementation and use of knowledge work support systems. On a practical level, our recommendations assist KMS developers in attempts to bridge the knowing-doing gap in organizations where individual members do not know or know of each other and the organization as a whole does not know what it knows.

The structure of the paper is as follows. In the second section, a theoretical background covering characteristics of organizational knowledge application, related IS research on system support for knowledge work, and Grudin's challenges for groupware developers is outlined. Thereafter, we present the research site and describe the method used. This is followed by a presentation of the KMS included in our study. Then we outline empirical experiences from three implemented and evaluated KMS. Using Grudin's groupware challenges for analyzing why the systems studied failed, section six develops our recommendations for how to integrate KMS with everyday knowledge work. In the final section, we discuss the implications of our work for IS research and practice.

THEORETICAL BACKGROUND

Formulated by researchers like Grant (1996), Nonaka and Takeuchi (1995), Spender (1996), and Tsoukas (1996), the knowledge-based theory of the firm postulates that services rendered by knowledge resources such as organization culture and identity, routines, policies, systems, documents, and individual employees form the basis for achieving competitive advantage.

Viewing the firm as an institution for knowledge application, however, Grant (1996) emphasizes that the competitiveness of an organization depends on its ability to effectively apply the existing knowledge and to take action rather than on the existing knowledge per se. Consistent with all theories of the firm acknowledging the efficiency gains of specialization, Grant suggests that the principal task of the organization is to coordinate the efforts of many specialists. In this way, organizational capability can be seen as the outcome of integration of specialized knowledge of multiple individuals.

Discussing fundamental mechanisms for integrating knowledge to create organizational capability, Grant argues that reliance on high-interaction and non-standardized solutions increase as task complexity and uncertainty grows. In such situations, problem solving relies less on organizational members following specifications and organizational routines and more on group efforts involving individuals with prerequisite knowledge and specialty. Distributed, unusual, and unstructured tasks and work processes requiring such personal and communication-intensive forms of integration can be described as characterized by variety

rather than routine and problematic to describe in manuals, job descriptions, and charts (Brown & Duguid, 1991). Typically, this type of work is performed by professional or technical workers with a high level of skill and expertise, for example, researchers, product developers, advertisers, and consultants.

Unlike service work, knowledge work defies routinization and requires the use of creativity in order to produce idiosyncratic and esoteric knowledge (Blackler, 1995). Knowledge work is thus untidy in comparison with operational or administrative business processes, in which tangible inputs are acted on in some predictable, structured way and converted into outputs. The inputs and outputs of knowledge work, that is, ideas, interruptions, or inspirations, are often less tangible, and in knowledge work there are no predetermined task sequences that, if correctly executed, guarantee the desired outcome (Boland & Tenkasi, 1995; Davenport, Jarvenpaa, & Beers, 1996). Summarizing the characteristics of a knowledge work process, Markus et al. (2002, p. 184) define such a process as an "organizational activity pattern characterized by (1) an emergent process of deliberations with no best structure or sequence, (2) an actor set that is unpredictable in terms of job roles or prior knowledge, and (3) knowledge requirements for general and specific distributed expertise."

Recognizing that knowledge work processes differ qualitatively from semi-structured decision making processes, Markus et al. (2002) argue that existing types of systems and their associated design theories do not adequately serve the unique requirements of this class of design situations. More specifically, they assert that the development literature on decision support systems, executive information systems, expert systems, organizational communication systems, organizational knowledge repository systems, and organizational memory systems does not provide sufficient guidance for how to build systems that support knowledge work processes.

According to Markus et al. (2002), the poor fit between the requirements of such work processes and existing IS design theories stems from three disconnects. First, decision-making in knowledge work processes requires that expert knowledge is adapted or contextualized to specific local conditions. Intended to support semi-structured decision-making, decision support systems and executive information systems do not provide system features handling expert knowledge and contextualizing translation rules. Resulting from this, these types of systems inhibit creative problem finding and solution generation. While expert systems manage general expert knowledge, they fail to support contextual knowledge and the flexibility needed for process emergence. Second, decision support systems, executive information systems, expert systems, and organizational memory systems are all specifically designed for a known type of user. Being designed for a particular type of user community, however, these systems are not well adapted to emergent work processes characterized by shifting user types having varying knowledge requirements. Third, today knowledge workers have access to many different types of systems such as decision support systems, expert systems, executive information systems, organizational communication systems, and organizational knowledge repositories. Since these systems often are isolated and not integrated into work practice, knowledge workers tend to manage their systems rather than getting the job done.

Arguing that a new IS design theory for systems supporting knowledge work processes is needed, Markus et al. (2002) developed a theory intended to assist systems developers in their efforts to design effective knowledge work support systems. On the basis of characteristics of knowledge work processes and requirements for information technology support of such processes, this theory matches principles guiding the selection of system features and principles guiding the development process with the unique user requirements of knowledge work. Elaborating on the theory developed, they suggest a set of additional research challenges. One concern is about how

Table 1. Eight challenges for groupware developers (Grudin, 1994)

1. Disparity in work and benefit. A groupware application typically requires extra work from individuals who do not perceive a direct benefit from using the application.
2. Critical mass and prisoner's dilemma problem. A group support system may not attract the critical mass of users needed to be useful or can fail because it is never to any one individual's advantage to use it.
3. Disruption of social processes. A groupware application can render activity that violates social norms, threatens political structures, or otherwise demotivates users critical to its success.
4. Exception handling. A group support system may fail to offer the wide range of exception handling and improvisation characterizing everyday group activity.
5. Unobtrusive accessibility. Support features for group processes are used rather infrequently, requiring unobtrusive accessibility and integration with more heavily used features.
6. Difficulty of evaluation. The problem of identifying and generalizing the factors underlying success or failure hampers learning from experience in the context of groupware development.
7. Failure of intuition. Intuition fails when the intricate demands on a groupware application are ignored, resulting in bad management and an error-prone design process.
8. The adoption process. Implementation of groupware in the workplace posits adoption challenges that go beyond past experiences of both product developers and large-scale information systems developers.

to keep KMS alive — updated, current, maintained — by encouraging use.

Even though KMS differ in significant ways from CSCW or groupware systems, we believe there are analogies suggesting that there are lessons to be learned from importing Grudin's findings to the KMS realm. Grudin (1994) presents eight challenges for developers of groupware applications that we argue are productive for achieving updated, current, and maintained knowledge work support systems (see Table 1). On a general level, Grudin's eight challenges call for better understanding of characteristics of work environments and for corresponding adjustments by systems developers. Whereas progress on the first five challenges requires better understanding of the requirements of the intended users' workplace, the final three require changes in the development process.

For the purpose of contributing recommendations for how to integrate KMS with everyday knowledge work, we shall use Grudin's (1994) eight challenges for groupware developers as a theoretical lens to analyze our three case systems. Targeting the KMS maintenance challenge, our discussion of the relationship between the recommendations developed and the three disconnects as identified by Markus et al. (2002) extends existing IS research on the implementation and use of knowledge work support systems.

METHOD

This work was carried out at Volvo Information Technology's (VIT) head office in Göteborg, Sweden, from August 1998 to December 2000. Employing some 4,300 people, and with offices in Sweden, Belgium, Brazil, Great Britain, Malaysia, and USA, VIT is today a rather large IT consultant firm and the Volvo Group's resource and expertise centre for IT systems. The main objective of VIT is to create global IT systems that generate value for their customers. Historically, VIT has achieved this by developing cost-effective systems where a significant percentage of the solutions were the same for the entire Volvo Group. A high degree of standardization was thus hailed as the optimal situation, and VIT's centralized mainframe operation, which had received several international awards for high efficiency and cost-effectiveness, had always been one of the cornerstones. By routinizing as much of the work as possible, VIT intended to ensure predictability, consistency, and quality in its services.

However, VIT was not the exclusive provider of IT services, because the companies within the Volvo Group also could purchase IT services from external providers if they so desired. But as long as mainframe processing was the core of the business, VIT was on top of the competition. The shift in the 1990s toward more Web-enabled solutions, however, opened the field for new, smaller, and quicker players. This situation put new demands on VIT's ability to change and adapt to new business solutions, and since then VIT has evolved from a Volvo internal resource and expertise centre for IT solutions to a global player serving customers also outside the Volvo Group.

The continuous development of knowledge, expertise, and skills needed for mobile services and IT in vehicles (telematics) is essential for VIT to continue to be a competitive partner in the future. The organization has therefore become, in part, more project-oriented and decentralized, and in such a situation empowering the employees to act more quickly and autonomously is important. The more rapidly changing environment and the more frequent exposure to previously unknown problem areas has resulted in a learning-by-doing situation rather than an attend-a-course approach to competence development. Skills are thus acquired and disposed of at a more rapid pace than earlier and, like many large organizations, VIT has recognized the problem of knowing who within the organization knows what. In an attempt to tackle this problem, VIT has initiated a number of initiatives over the last few years to reinforce its knowledge management process: creation of homepages for projects, groups, and departments, establishment of human networks related to particular competence and knowledge areas, evaluation of search engines and agent technology for the intranet, implementation of IT support for managing competencies, knowledge, and resources, and development of trainee and management programs. In this paper, we report experiences from our involvement in VIT's efforts to implement and test support systems for knowledge work.

We think it is fair to describe this work as a case study. Yin (1994, p. 13) describes a case study as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident." This definition captures eloquently the characteristics of our study of the role of support systems in knowledge work. Pointing to analytical differences and considerations, the literature distinguishes between multiple case studies and single case studies (Yin, 1994). However, in practical work the boundaries can become blurred. As asserted by Yin, reflecting the rationale behind multiple experiments, multiple case studies are typically carried out as to provide replication and thus be regarded as more robust. Following this, the cases must be selected to produce either literal replication (similar results are predicted) or theoretical replication (different results but for predictable reasons) (Yin, 1994).

The work described in this paper has emerged out of three interrelated projects at VIT over a period of 30 months. Each of these projects had its own research agenda, its own research questions, and produced its own output (e.g., Lindgren & Stenmark, 2002; Lindgren Stenmark, & Ljungberg, 2003, 2004; Stenmark, 2001). What we present in this paper is a post hoc analysis of the entire process, where we have revisited the original data and looked at it through a different theoretical lens. Although it is obvious that our work covers multiple cases, we do not claim it to be a multiple case study by Yin's definition because the objective has never been replication. Instead, we like to think of our research effort as a continuous study of a theme (system support for knowledge workers) that involves three interrelated cases, each providing distinct pieces to a collective puzzle. While Yin acknowledges that case studies can be used both qualitatively and quantitatively, it is evident from his text (the use of hypothesis to test, the concern for internal and external validity, and the issues of reliability) that he comes from a quantitative tradition. In comparison, we are more qualitatively

oriented and our understanding of a case study is thus more in line with Walsham's (1995) description of the interpretive case study.

During the first project, referred to as the Watson case, the first author spent four months implementing an agent-based recommender system and studying its adoption and use. For this study, approximately 80 users were invited, of which 48 agreed to participate. The objective of the research, the concept of agent-based systems, the design rationale of the application, how to operate the system, how to register and login, and how to set up and run individual agents were explained to all participants at a two-hour introduction meeting (in total, three such meetings were held). User experiences as well as hard data were collected in several ways, including interviews, questionnaires, and Web server log file analyses. First, all users were invited to a group interview/focus group session, but only eight showed up. The remaining 40 users were then sent an e-mail questionnaire, which only 12 respondents answered. We thereafter conducted seven semi-structured and open-ended interviews, each lasting between 28 and 66 minutes.

The second project, here called the Tieto Persona/Human Resource (TP/HR) case, was a joint venture between the researchers and VIT practitioners. While VIT personnel did the actual coding, we as researchers were allowed to provide input to the implementation process. User viewpoints from the TP/HR system were collected through 10 semi-structured interviews, which lasted between 45 and 60 minutes. The interviewees were selected to represent different organizational roles and positions, including management consultants, systems programmers, and personnel from the human resource (HR) department. Another important source of TP/HR data was archival records and project documentation (covering strategy plans for knowledge/competence management in VIT and written material about technical aspects). Being members of the TP/HR project team allowed us access to that kind of material.

The third project — the VIP case — was conducted on our initiative and again carried out as a collaborative effort involving researchers and VIT practitioners. However, the prototype was developed and implemented by our research team without active involvement of VIT members. Simply put, VIT had too many resources tied up in the TP/HR project to simultaneously engage in yet another development and implementation project. In order to gather empirical data, we conducted 16 semi-structured one-hour interviews with VIP users. The interviewees again occupied different positions within the organization, ranging from non-technicians such as HR staff members, project managers, department managers, and financial controllers to technology watchers and systems programmers (many of whom had also tested the TP/HR system).

As is evident from the previous discussion, this work stretches over several years and across multiple projects, resulting in the use of a variety of research methods and techniques. User experiences and hard data have been collected in several ways and from multiple sources, including interviews, focus groups, questionnaires, archival records, system documentation, intranet documents, and Web server log files. Two of the projects included elements of action research where we collaborated with the practitioner at the researched site. The first author was actually employed by the researched organization at the time of the three projects.

When we reflect upon our research, in retrospect, we are able to notice patterns that eluded us whilst being in the midst of things. Indeed, the three cases together allow us to draw conclusions not possible from the individual cases. Miles and Huberman (1984) describe the analytic phase as consisting of three concurrent flows of activity: data reduction, data display, and conclusion drawing. Data reduction refers to the process of selecting, simplifying, and abstracting the raw data that the researcher has in forms of field notes and transcribed interviews. Data reduction is part of the analysis because the researcher makes explicit choices of what categories to use, what sources to include, and what data to summarize. Data display is an organized spatial way of

presenting the data systematically to the researcher in a form that helps the researcher see what is happening. Again, the process of data display is part of the analysis, since deciding what data to present and in what form affects the result space. Conclusion drawing, finally, is to decide what things mean. This process is also part of the analysis. Conclusions first appear as vague hunches that subsequently are validated as the work proceeds. Qualitative data analysis is thus an iterative enterprise that also includes the data collection phase.

In our work, we have certainly made use of data reduction, data display, and conclusion drawing. However, since all data was collected as part of previous research projects, these activities have been separated from data collection. Ultimately, this means that we have been limited to data already collected (and to some extent also reduced and displayed). In terms of data analysis, this may have biased our understanding of the case. However, we have transcended the individual cases in retrospect by focusing on the learning process that has developed over the years and across the cases. In addition, we have also applied new theory to reinterpret the data.

The theoretical framework that has guided our analysis is based on Grudin's (1994) eight challenges for developers. During the analytic phases, the data has been read, categorized, conceptualized, and interpreted in an iterative fashion. Having data not only from one prototype but also from three prototypes in sequence, we have been able to follow an over-time development for each of the themes suggested by Grudin. The three cases have been compared with and contrasted to each other and with Grudin's themes in an iterative fashion that has furthered our understanding of the phenomena under study.

SYSTEM DESCRIPTIONS

Our first case is the development and evaluation of the Watson (*What's on*) prototype. The Watson prototype project was initiated in August of 1998. VIT's intranet, which had been in place since 1995, was growing quickly both in terms of content, servers, and users, and in 1998 it consisted of some 450 Web servers hosting a little under half a million documents. As a response to a rapidly growing intranet, our basic idea behind Watson was to examine how an agent-based recommender system could help knowledge workers to deal with information overload by providing awareness of relevant intranet information.

Watson was built on top of Autonomy's AgentWare software, which is a commercially available tool that uses neural networks and advanced pattern-matching techniques to find similarities between texts. Watson indexed VIT's intranet each night and synthesized each Web document to a 0.5K digital representation, that is, a "fingerprint". Once fingerprints had been created, the system's reasoning engine could perform concept matching, that is, finding documents with similar fingerprints. New users had to create a user profile in which they could describe their job role or work responsibilities in a free text fashion. If a user already had a curriculum vitae (CV) stored elsewhere, it could be copied into this field. Once saved and stored, the user profile was then converted to a digital fingerprint. The user could thereafter set up and train an agent to look for information on a specified topic. This task corresponded to submitting a search engine query but was expressed in natural language. Indeed, the best results were achieved when users pasted a large chunk of text from a known, relevant document and asked the agent to find more similar documents. The agent profile would then be converted to a fingerprint and compared to all other fingerprints in the system. The returned results could be inspected and the user could give the agent positive feedback when highly relevant documents had been found, thus further tuning the agent profile and enhancing the matching capabilities. The profiles used in the search could therefore be said to be implicit.

The rationale behind using software agents was to off-load VIT-members from having to search the intranet themselves, thus providing users with a direct and personal incentive to use the prototype. Since documents,

user profiles, and agents were all represented as fingerprints, the system could be used to search for things other than pure information.

First, we implemented a community feature that was intended to enable knowledge workers to locate colleagues with similar assignments and organizational roles by matching their job profiles. Clicking the community button, the user's profile fingerprint would be compared to other users' profiles, and a list of users with matching profiles was displayed. In this way, the user would get the name and contact information of all employees with similar job roles. The intention with this feature was to make users aware of each other's presence and thus facilitate the emergence of online communities. Second, the *Similar Agents Feature* worked much like the community feature, except that it was not the job profiles that were matched but the agent profiles. Initially, this was meant to allow users to find similar agents in order to have them cloned. In this way, new and inexperienced users would receive help to get their agents to a decent quality level more quickly. However, the cloning service was not implemented in time for the study. The only feature offered to the users during the test was the option to find other users with similar agents.

The Watson prototype was tested locally without top management support. Although user reactions were positive, the technology was considered too expensive (at the local level) and the prototype was not further developed.

The second case concerns our involvement in the implementation and evaluation of the TP/HR system. Explicitly targeting management of core competence and skills, the TP/HR project had three key objectives: (1) to identify and construct a competence structure that could serve as a foundation for mapping the employees' expertise and knowledge, (2) to implement the competence structure in the TP/HR system, and (3) to develop and establish an accompanying maintenance control function for keeping the system structure updated and relevant.

TP/HR was a commercial, off-the-shelf module-based client/server system that was implemented in February 2000 through a top-down strategy where the competence structure was defined by management alone. In VIT's implementation of TP/HR, competence was divided into functional (i.e., tasks such as project management) and technical skills (e.g., java programming), which in turn had sub-levels ordered in a complex tree structure. This structure was the result of a multi-months process in which several of the companies in the Volvo Group had been involved. Despite the massive effort, the result was more complicated than originally anticipated and yet not perceived as optimal. Nonetheless, the employees were supposed to navigate the competence structure to find the individual competencies that applied to each user and then rate their competence level on a scale from 1 to 5. This data was the stored in the TP/HR system's database.

In order to preserve integrity, TP/HR did not allow knowledge workers to see each other's competence descriptions or search for particular expertise or skills. Such finding competence features were exclusively for managers who could search for and find employees holding a particular competence on a certain level, for example, a java programmer on level three or above. Managers could also invoke other fancy features such as competence gap analyses that would indicate any discrepancies between the aggregated competence level as recorded by the system and the estimated future need as calculated by management. VIT planned to use these competence analyses to support organizational activities such as resource and availability planning, internal and external recruiting, goal and personal development discussions, forming teams of employees, finding competence when manning assignments, and mission steering. In this way, TP/HR was intended to be a knowledge work support system for both short- and long-term objectives.

The TP/HR system was not only heavily promoted by VIT management but also by several other member companies in the Volvo Group. Despite the total cost of the system itself and the complicated process of constructing a com-

petence system ready to implement, the system was endorsed and tested. However, as a result of negative test experiences associated with user commitment and system maintenance, VIT decided to put the TP/HR project temporarily on hold.

The third case concerns the implementation and evaluation of the Volvo Information Portal (VIP). In the winter of 2000 (January to April), this recommender system prototype was developed as an attempt to tackle an information overload situation that was even more articulated than during the Watson study. Resulting from a rapid growth, the intranet had increased to more than 700 Web servers hosting close to 750,000 pages.

Similar to the Watson prototype, VIP was an agent-based recommender system built on Autonomy's AgentWare platform. VIP allowed knowledge workers in VIT to define information agents that searched an index database for intranet documents matching their interests. By defining one or more agents and providing each of these with an interest profile, VIP users were thus able to have the corporate intranet monitored for interesting items. From a user's point of view, the primary objective was to receive relevant and targeted information as effortlessly as possible. Therefore, it was in the users' own interests to define the interest areas as well as possible because a well-defined profile would reward the user with high-precision search results. On a general level, the features offered by the Watson and VIP prototypes were pretty much the same. However, two major changes were implemented in the VIP prototype.

First, the user profiles, that is, the part where users explicitly stated their job descriptions, were abandoned since the Watson study had told us that the explicit profiles were not perceived as useful. Second, Watson's similar agent feature, which allowed users to locate other employees interested in the same areas, was, in contrast, heavily used and appreciated. Building on this learning outcome, we introduced a find competence feature that allowed employees to find a fellow colleague with an arbitrary interest (not just people with interests similar to their own but any interest). A user wanting to find someone dealing with XML could use this feature to type in a few XML-related keywords and receive a list presenting all employees with active XML agents. Since ordinary knowledge workers had been excluded from this kind of search for knowledge and skills in TP/HR, this specific feature was intended to provoke a reaction from the organization.

As was the case with Watson, VIP was a local initiative and there was virtually no top-management support to promote this type of solution. The prototype was not further developed but some of the lessons learned were discussed in terms of implications for other knowledge work support systems in VIT. In particular, implications for systems supporting career management, recruitment and selection, and training were discussed.

Table 2 summarizes particularities of Watson, TP/HR, and VIP and design relationships between the three systems.

EMPIRICAL EXPERIENCES

Watson

When evaluating the Watson prototype, we soon realized that we had underestimated the difficulties involved in agent training. The users conceived setting up and training of agents as non-trivial, and many users had experienced mainly negative results. A majority of the users reported "strange" or "unexpected" document matches. However, the most interesting results came from the community and similar agent features.

The community feature was intended to enable knowledge workers with similar job profiles to learn of each other's existence. Not many users exploited this feature, though. Those who actually did try this feature used it only once or, in one case, twice. The low interest was not due to bugs or technical malfunctions, since most interviewees considered the community feature to be working, that is, it delivered what it was supposed to. Instead, the low interest was attributed to the fact that the result was

Table 2. Design relationships between the three systems

Systems	Watson	TP/HR	VIP
Duration	August-November 1998 (4 months)	July 1999-December 2000 (18 months)	January-August 2000 (8 months)
Technology	Agent software for information retrieval	Database for storage of competence data	Agent software for information retrieval
Motives for the organization	• Increased information awareness • Effective information management	• Systematic core competence mapping • Competence gap visualization capability	• Increased information awareness • Effective information management • Competence identification capability
Motives for individual knowledge workers	• More targeted information • Community building support	• Marketing of knowledge and skills	• More targeted information • Community building support • Expertise location support
System content	• Implicit profiles for information • Explicit profiles for community building	• Explicit competence descriptions	• Implicit profiles for both information and community building/expertise location
Level of support	Local	Central	Local
Outcome	The prototype was not further developed	VIT decided to put the TP/HR project temporarily on hold	The prototype was not further developed but some of the lessons learned were discussed in terms of implications for other knowledge work support systems

just not very exciting in that the users already knew the people doing similar jobs. Many users with similar profiles worked at the same departments and were not too interested in finding like-minded colleagues. One of our respondents put it as follows:

What's the use of hooking up with people doing the same stuff I do? If I want to talk to these guys, I go talk to them. They sit over there. But take, eh … databases — SQL server or something — where I don't have a clue. I wouldn't know where to start. It would probably be better to team up with those who know stuff I don't know.

As a substitute, the respondents suggested that one should be able to search for people with profiles other than one's self (because this was a design implication apparently shared by many users and one that seemed to be adding value, we implemented it in the VIP prototype). The low utilization of the community feature can be seen as an implicit critique of the underpinning principles of explicit profiles. One user actually explicitly complained about this specific feature, claiming to have been connected to people he did not know. This was not what he had expected, and he concluded that, "this was clearly a bug." While people are often viewed as performing their jobs in line with their formal job descriptions, the Watson evaluation provides evidence of the opposite. The community feature was built on static profiles created by the users themselves to mirror the official responsibilities placed upon them

by the organization. However, these profiles were not only already known to the members but also experienced as fictitious and depicting an espoused theory of work. A later inspection of the users' profiles suggested that they invested a minimum amount of time on these profiles. We found the profiles to be very short and sketchy, containing merely department name and job title.

Although the similar agents and the community features incorporated the same pattern matching mechanisms and generated exactly the same output, the former was much more frequently used. Several respondents reported that they were surprised to find certain people sharing their interests. The interviewees were also intrigued by the fact that the similar agents feature returned users whom they had not expected to be interested in a particular topic. One of our respondents said:

Sometimes you think you're alone and then you find out you're not. And it's not… I mean, it's all kinds of different people. It's really interesting to see who else is searching for these sorts of things.

Users clearly appreciated this opportunity to see in what areas other organizational members applied their knowledge, considering these results to be useful new insights.

TP/HR

Many of the participants in the TP/HR pilot project were positive about the system, which, in their opinion, was a first step toward some structure and order in an otherwise rather chaotic situation. Even though they complained about the old-fashioned user interface, they thought that TP/HR would be a useful tool, particularly in establishing a common terminology. An agreed common vocabulary helps make competence more tangible and thereby assists managers in both coaching dialogues with the employees and competence gap analyses. Updating the competence description should be a responsibility shared jointly by the manager and the employee. Typically the employee performed the physical input, closely assisted by, and in dialogue with, the manager. The competence description should be updated as often as possible to reflect developments since the last update. However, not only did the employees' knowledge and skills change frequently, but the competence structure itself did not remain correct for long. While entirely new competencies made their appearance, existing knowledge and skills became obsolete much faster than the TP/HR system was designed to handle. A management consultant stated:

Earlier it was easier [to have an updated system], since there were few programming languages. Now the development is so fast. Yes, there are the fourth, fifth, and sixth generation.

To cope with this evolution, VIT established a maintenance organization. Keeping the system structure and the competence data up-to-date was a burdensome task, requiring a lot of administration, though. As the project proceeded negative aspects started to surface. It seemed that knowledge workers at the grassroots level had no direct interest in providing information about their skills since they could not benefit from using the system. A management consultant pointed out:

TP/HR is hierarchically structured and closed. As an individual, you can see nobody but yourself. If I search for a certain competence, the system should support me in identifying the appropriate person. Such features are missing in the system. Instead, I have to talk to someone who is familiar with the employees' knowledge and skills. In any case, I can't use the TP/HR system for doing it myself.

Despite the intended change toward a more project-oriented and decentralized organization, VIT's organizational structure can be described as hierarchical. This was reflected in TP/HR's closed system structure. While managers were authorized to see information about all their subordinates, employees in other posi-

tions could only access their own descriptions. However, during the initial phase of the TP/HR pilot project, the ability to search for and find a person with specific skills was considered an obvious feature. As the pilot project advanced this changed, making the TP/HR be primarily a management vehicle including features for measuring the status of employees' competencies and gap analyses. The employees were presumed to regularly feed the system with competence information, but they did not get much in return. As highlighted by several respondents, this producer/consumer dilemma counteracted the employees' motivation to use TP/HR. During the evaluation, the interviewees discussed different motives as to why VIT had chosen to implement a system with a closed structure. One TP/HR project group member pointed out the following reason:

The more people involved in competence registration, the more regulations there must be. We don't want other managers to be able to conduct internal recruiting [by using TP/HR].

In line with this quotation, several project group members discussed TP/HR's closed system structure in terms of a means for avoiding internal recruiting in the organization. Many of these respondents argued that TP/HR could have been an important tool for employees to communicate their existing skills and ambitions for future development. A management consultant gave her opinion of the matter:

To use the system would be a way to market your self to get interesting assignments. The opponents to this argument are surely those ten percent who have come to a stand still in their competence development. Presumably, there are many mangers in this group.

However, the closed system structure conveyed that competence was primarily a personal thing of no interest to others. Furthermore, several respondents highlighted that TP/HR lacked features that handled information about employees' wanted skills and desired work tasks. Project management members did not seem to think that this was much of an issue, though. The TP/HR project manager claimed:

Interests are a long way down on the list. It's fundamentally a personal thing; interests have no strategic value according to my point of view. Interest is for your own sake, and therefore it's not reasonable to assume that people should register this type of information in the system. […] People won't invest their time in such work because they simply don't benefit from it.

However, some project members did not fully appreciate this standpoint, as they saw interest as an important dimension of knowledge work. A project manager involved in the system implementation said this when discussing the rationale of TP/HR:

It's important that we're able to find and take care of people's interests. Definitely you perform better if you are interested in the work-task in question. And surely the employees' potential to learn increases when they find the actual area exciting.

As this quote illustrates, there were people who had different perspectives as to what type of information that should be handled by the TP/HR system.

VIP

The system evaluation indicated that the interviewees viewed VIP and its content in different ways. Some users thought that VIP contained formal descriptions of skills and knowledge in a similar manner to TP/HR, while other respondents were uncertain as to what type of information VIP handled. This ambiguity is illustrated by the following quote from a software developer:

Well, the find competence feature; first I interpreted it as if you came to some kind of

competence/skills database. There's one competence database that I subscribe to where you search for skills. For example, if someone knows C++ and COBOL and what have you, then you can search for it. So, it does not seem intuitive that this is called find competence, but maybe it's right. I guess it's something you have to get used to if you want to use it. But it does not seem intuitive […] I'm still puzzled when I look at it.

However, the majority of the interviewees were rather attracted to the fact that VIP managed a different type of information than TP/HR. VIP was based on people's everyday actions in the form of information seeking activities, and several respondents saw the system's potential to present an updated picture of the organization's knowledge and skills. Pointing to its integrated character, one HR manager argued that VIP could indicate what people actually use their skills for:

TP/HR is a lot about order and being in control of the situation; to know what we have and the level of education of our employees… how many of these and how many of those. Then this prototype is something else. It is what people do on an everyday basis. It is what they used their skills for. It is sort of the next step.

According to many interviewees, VIP can provide VIT information about knowledge and skills that are applied in the organization. Also, VIP makes it possible to identify people searching for information outside their formal area of responsibility. As highlighted by the respondents, such actions typically indicate a natural driving force. The fact that VIP was a system with the ability to visualize people's commitment and interests attracted most of the participants in the system evaluation. Indeed, one of these was the manager of the TP/HR project:

It is interesting to be able to find colleagues who are interested in the same things. Because our main problem here is that there are people working with similar things everywhere and you don't really find them. For me, it was natural to see the other users but also to signal my own presence and interests to them.

As people added, deleted, or retrained their agents, these unnamed communities would constantly change members to reflect the current situation and the actions of the users themselves. No organizationally appointed administrator had to define communities in advance according to some espoused theory; the organizational members were instead in control. A software developer familiar with both information retrieval tools and the TP/HR system commented:

The advantage with this approach is who controls it, I guess. In a conventional system, the administrator measures the information and controls it, and builds the system himself. Here, as a user, I'm able to influence the result to a much higher degree. This system [VIP] is built on organizational needs. By using this system, I can affect my situation by expressing my wishes. I want to work with XML, for instance, although I don't do this in the present situation.

The VIP system was based on the intentions and actions of individuals. Discussing VIP in terms of a decentralized system where the users themselves to a large extent affected and decided upon the content, several interviewees associated the system with development, change, and learning.

RECOMMENDATIONS

Disparity in Work and Benefit

The fact that groupware applications expected to provide a collective benefit still means that some people will have to adjust more than others and is probably something most groupware users have experienced. In other words, such applications often require additional work from individuals who do not directly benefit from using them (Grudin, 1994). As is evident from our Watson account, knowledge

workers were expected to supply their own profile descriptions for enabling other users to find them when engaging the community feature. The profiles thus had to be created for someone else's benefit, resulting in predictable and uninteresting descriptions. The TP/HR system was based on a similar design rationale. Employees were supposed to create and maintain their own competence database entries without even being able to use the system. The expected benefit was on an organizational level only. To tackle this type of problem, Grudin suggests that making the additional work required someone's explicit job might be a workaround. Such a solution seems appropriate when large organizational KMS are involved and associated management incentives are present. Another approach perhaps more feasible is to design KMS with an accompanying process ensuring that usage creates tangible benefits for all key actors involved. Indeed, striking the right balance between cost/benefit for a multitude of actors is a challenge in itself. Targeting situations where extra effort is needed from knowledge workers interacting with the KMS, our first recommendation is that knowledge work support systems must result in perceived value to reinforce user commitment.

Critical Mass and Prisoner's Dilemma

A groupware application requires a high percentage of all group members to interact with it to be truly useful. Depending on individual role or status, one or two defections may be enough to thwart an otherwise successful deployment. The problem is often to induce early adopters to stay on and not abandon the tool until a critical mass of users is achieved and they all can start to benefit (Grudin, 1994). With only 50 or so users in the Watson case and approximately 30 in the VIP study, there were significant risks that individual users would create agents for which there were no matches. The community feature would in such cases result in zero hits, thus generating no additional value. Since the primary incentive for signing up with the applications was not to find community members but to receive targeted information as a result of training an agent (the more accurate agents, the better results), the lack of community members may not have had a negative impact on the overall use. Grudin argues that management can force a critical mass by removing alternatives or mandate system usage until users experience benefits and thus voluntarily continue to use it. While this was the intended strategy in the TP/HR case, the benefits were never planned to occur on the individual level, and reaching a critical mass did not help the system to survive. In contrast, the VIP system provided every individual user with targeted information. The incentive to participate was already there, and a critical mass was not required to receive the primary benefits. Our second recommendation for developers of knowledge work support systems is thus to lower system thresholds by minimizing the amount of additional work required and to build in incentives for use by making salient both individual and collective benefits.

Disruption of Social Processes

Group activities are highly dependant on implicit social, motivational, economic, and political factors that change over time. Developers of groupware applications ignoring such critical factors may inscribe behavior in their tools that is at odds with the subtle social dynamics of the organization. If the tools violate social taboos, upset existing power structures, or reduce financial motivation, organizational members are likely to put up resistance (Grudin, 1994). Since knowledge and skills are increasingly valuable resources in modern organizations, one can expect knowledge workers to be reluctant to make explicit their knowledge and allow it to be captured by some KMS for the good of the collective. Such a process may result in them losing not only power and money but ultimately their jobs. TP/HR was clearly a top-down system designed for managers, supporting a management perspective. Moreover, to avoid internal recruiting of experts and key individuals, the TP/HR system was closed to all but senior managers, thus effectively removing the possibility for individuals to market

themselves. As our empirical experiences indicate, social factors with high influence on a grass-roots level were not considered. When discussing possible solutions, Grudin reminds us of the importance of avoiding the assumption that work is carried out in a "rational" fashion. Obviously, while some rationality is involved, everyday work has typically more to do with individual actors' hidden agendas than with some agreed-upon organizational goal. In the context of knowledge work practices, workers may be reluctant to make explicit their knowledge because they fear losing power and competitive advantage. In contrast, knowledge can also be seen as an infinite resource that is not reduced when shared. Indeed, which interpretation prevails is highly situational. For knowledge work support systems to be successful in organizations, our third recommendation is therefore that such systems must acknowledge and coexist alongside existing cultural and social processes.

Exception Handling

When groupware applications are designed and implemented based on official office work handbooks and other readily available work specifications, the resulting tools may end up supporting the way things are supposed to work (rather than the way they do work). Realizing that descriptions of standard procedure often are post hoc rationalizations, we may recognize that what makes possible efficient performance is the ad hoc problem solving capacity of man (Grudin, 1994). For good reasons, the industrial organization has been preoccupied with structures and standards. However, the breakdown of bureaucracy occurs when exceptions start to outnumber the routine. When yesterday's knowledge is no longer a prerequisite for tomorrow's work, old knowledge does not only become obsolete, it may even be harmful to the organization. Obviously, knowledge must be renewed and find novel paths continuously to remain valuable. As we saw previously, TP/HR was implemented based on formal work manuals and corporate strategy policies. Resulting from this, many of the skills needed and work situations encountered during an ordinary office day were not covered by the system. Rather than supporting rational myths, Grudin argues, we must carefully study how work is actually done. Because exception handling and ad hoc problem solving are the birthplaces of knowledge, KMS without the ability to facilitate these situations have less potential. Needless to say, KMS must be tailorable and provide flexibility, although these requirements present challenges in themselves. As knowledge workers apply their experiences and skills in innovative ways to handle unstructured tasks, our fourth recommendation is thus that developers of knowledge work support systems should seek solutions with the capacity to leverage these everyday activities.

Unobtrusive Accessibility

Even in groupware applications, the bulk of the work is carried out as individual tasks performed by individual group members who mainly use groupware features to coordinate and communicate the result. As a consequence, groupware features are typically used less frequently than many of the features supporting individual activities (Grudin, 1994). Building on this observation, Grudin asserts that less frequently used feature must be tightly integrated with features that most users engage to catch on. In addition, such integration must be unobtrusive not to obstruct the use of the more frequently used features. Except for once in a while using the system for updating his or her profile to comply with corporate policy, the individual VIT employee had no reason to enter TP/HR. In contrast, VIP rewarded users by serving targeted information and monitoring the indicated fields of interest on their behalf. Assuming information handling to be something organizational members engage in on a day-to-day basis, information agents would probably be a welcomed and relatively often used resource. The profiles derived from agent usage would then be maintained both frequently and unobtrusively. Striking the right balance between being unobtrusive and yet accessible is otherwise indeed a challenge. Grudin suggests that

infrequently used features should be added to and incorporated in existing and already successful applications rather than being launched as separate systems. With such an approach, Grudin argues, the system can over time educate the users and slowly make them aware of beneficial spin-offs. In terms of implications for KMS, such systems must not be introduced as explicit stand-alone applications that knowledge workers intentionally must interact with in addition to their other job responsibilities. Our fifth recommendation is therefore that knowledge work support systems should instead be invoked when knowledge is applied in practice by exploiting spin-offs from activities that knowledge workers are already engaged in.

Difficulty of Evaluation

Whereas interaction with single-user applications can be sufficiently covered during an hour's observation, groupware interactions involve many different users and unfold over much longer periods of time. This makes evaluation of groupware applications more complex and less precise. While determining whether an application is a success or a failure may be easy, it is more difficult to identify the factor(s) responsible for the result (Grudin, 1994). We were able to evaluate the VIP prototype by studying single users attending the primary objective of receiving relevant corporate information. However, we were less successful in evaluating the organizational impact of the system, since such an evaluation would have required a much larger test population. The lack of historical data and ephemeral nature of the implicit profiles further added to the difficulties. However, TP/HR was even more difficult to correctly evaluate. Obviously, only three explicit competence profiles would have been a failure, but the existence of 30,000 profiles would not necessarily have indicated success. The question here is whether the organization or the individual should decide if a KMS is successful. As argued previously, there must be a benefit on the individual level before there can be a positive organizational effect. Yet if return on investment is noticeable only at the individual's level, organizational sponsors may decide to abandon the system in lack of tangible proofs of success. Grudin's advice for how to deal with the problem of evaluation is to ensure the right mix of skills, that is, technical, sociological, and organizational, are allocated for the development task and to disseminate the results actively to all stakeholders. His experience from the CSCW community is that too little accumulated learning is taking place due to the inability to learn from experiences. An important note in the context of *in situ* KMS evaluations is that knowledge is an intangible resource typically affecting both individuals and the organization as a whole indirectly. Ultimately, this means that it is very difficult to isolate the single factor contributing to the result. It may in fact not be one single contributing factor but a chain of concurring factors. The inherent nature of knowledge itself thus makes evaluations of KMS even more complicated. Responding to this challenge, our sixth recommendation is that KMS evaluations must involve different knowledge worker groups and be designed as collaborative efforts targeting individual benefits as well as organizational effects.

Failure of Intuition

When software is constructed by the same people who are going to use it, intuition can be a reliable input to the design process–at least as far as single-user applications are concerned. Whereas most organizational members have informed ideas about what is required to get the job done, individual intuition is less likely to be able to predict the intricate demands on groupware tools that are to be used by a wide range of different users. In many cases, the unwelcome extra work required of other users to get the application to work is underestimated (Grudin, 1994). Developers typically rely on feedback from a few potential users (or sponsors), and it is often these actors who are expected to benefit the most. A parallel from the VIP study is the TP/HR system where mostly HR staff and managers (typical stakeholders)

were involved in the evaluation. In contrast, Watson and VIP were designed by a knowledge worker for other knowledge workers. An interesting observation is that managers on average were less impressed with the VIP approach than were ordinary employees. According to Grudin, relying less on (stakeholders') intuition and more on user participation is the way forward. This may lead to fewer projects being run but hopefully also to more realistic design goals and higher success rate amongst those that are actually started. Reflecting this argument, systems designers capable of identifying managers' needs should be engaged when building KMS to support management, while entirely different developers should be brought in when designing for another user group. Our seventh recommendation therefore suggests that there should probably not be one large knowledge work support system solving everything but rather many small applications handling more specific aspects of knowledge management.

Adoption Processes

Due to the critical mass problem mentioned earlier, groupware applications require more careful introduction in the workplace than developers may appreciate. Hence, they must pay more attention to the adoption process than product developers have in the past. The lower visibility of groupware features, which in turn generates less management support, also means that groupware developers face more difficult acceptance problems than large-scale information systems developers (Grudin, 1994). In our field studies, we noticed how the number of volunteering test users decreased from Watson to VIP, which is something that typically happens when the group's curiosity wanes and people's attention returns to their ordinary work. Although our KMS prototypes were based on information seeking — a process most employees are familiar with — the tools themselves were new and unknown and obviously suffered from adoption problems. The small scale of our project, and consequently limited managerial attention, is likely to have contributed to the death of the prototypes. Grudin's solution to this problem is to sidestep the introduction problem as much as possible by adding features to existing applications, as discussed earlier. Building on the success of established systems and functions would, if not guarantee, at least substantially increase the likelihood of survival. As people continue to use the system, they will eventually discover the benefits of the added features, and system usage will be further reinforced. Our eighth and final recommendation is that in situations where KMS depend on input from and interaction with many knowledge workers in the organization, familiar applications used by many employees (*e.g.*, e-mail applications, word processors, Web browsers, or printer spooling systems) should be selected as hosts for the knowledge management features to be added.

DISCUSSION

The knowledge-based theory of the firm postulates that knowledge is a key to the continued vitality of organizations (Grant, 1996; Nonaka and Takeuchi, 1995; Spender, 1996; Tsoukas, 1996). However, whereas knowledge in organizations has the potential to be applied across time and space to yield increasing returns, managing knowledge as an organization-wide resource as to facilitate its application is not easy (Garud & Kumaraswamy, 2005). Seeking ways to reduce the knowledge gap, organizations are attempting to leverage their knowledge resources by employing various forms of knowledge work support systems (Kankanhalli et al., 2005). For example, such support systems can enhance application of knowledge by facilitating the capture, updating, and accessibility of organizational information and knowledge (Mao & Benbasat, 1998). Needless to say, while technological capabilities are critical, having sophisticated support systems does not guarantee success in knowledge management initiatives (Kankanhalli et al., 2005). Indeed, although the rationale for investing in KMS supporting knowledge application may seem convincing in theory, such systems tend to fail when implemented in the everyday practice of

contemporary organizations (Schultze & Boland, 2000). Following this, the development of systems with the capacity to bridge the knowing-doing gap in organizations has been recognized as a significant area of KMS research (Alavi & Leidner, 2001). In this context, as asserted by Markus et al. (2002), an important challenge concerns how to keep KMS alive—updated, current, maintained—by encouraging use.

This paper reports empirical experiences from three implemented and evaluated KMS at VIT. Our evaluation of the three KMS revealed a number of insights that we believe have both theoretical and practical significance for the development of knowledge work support systems. An important finding from our involvement in VIT's attempts to implement and test support systems for knowledge work is that knowledge applied in practice was what attracted organizational members. As an illustration, the Watson study clearly showed that profiles based on practice are considered more trustworthy than descriptions based on espoused theory. Indeed, several respondents highlighted that the prototype not only supported individual knowledge workers in their everyday actions but also the organization as a whole because valuable knowledge resources could be identified and used more effectively. This finding suggests that knowledge work support systems need capabilities that cater for ongoing actions of organizational members as they sought to apply the knowledge necessary to perform their day-to-day tasks. As noted in the literature, however, to support what knowledge workers actually do by making authentic work activities the primary focus of KMS implementations requires a thorough understanding of both the tasks and their performance of the tasks (Burstein & Linger, 2003).

Furthermore, when investigating the underlying reasons for the problematic introduction of TP/HR, we came to realize that the closed system structure in combination with the system's lack of future orientation negatively affected knowledge workers' willingness to use and contribute to it. Clearly, while TP/HR offered few use incentives on behalf of knowledge workers at the grass-roots level, the system added to the users' workload and obliged them to do things in addition to what their tasks at hand required. This finding verifies earlier research suggesting that knowledge work support systems must support various forms of employee incentives (such as enjoyment in helping others, better work assignments, and promotion) as to encourage usage and build a critical mass of users (Kankanhalli et al., 2005). Indeed, as recognized by Stein and Zwass (1995, p. 107) "a user's intrinsic motivation to contribute information to the system differs with the degree to which the contributed information is instrumental to the user's goals."

Building on the learning outcomes from Watson and TP/HR, the VIP prototype was specifically designed to test our idea that exploiting knowledge workers' everyday actions in an unobtrusive manner would actively afford user participation. According to our respondents, exploiting traces that knowledge workers' everyday activities leave behind in form of published documents and/or search queries is a promising way to reveal otherwise invisible patterns of knowledge application. In this way, they argued, the organization can begin to find expertise and skills as soon as knowledge workers start to apply their existing or emerging knowledge. In view of the fact that organizational routines for knowledge management often become so inflexible that they form the basis for stagnation (cf. Garud & Kumaraswamy, 2005), this finding suggests that organizations should seek dynamic knowledge work support systems capable of reducing the time and effort needed to capture, codify, and visualize knowledge. In this context, Kankanhalli et al. (2005) envision that future KMS will enjoy the capability to support the dynamism of knowledge work processes by allowing for more natural forms of knowledge contribution (e.g., audio or video) as opposed to purely text contributions.

Reflecting on these insights and the entire three-year research project, we have used important observations from the CSCW field

as a theoretical lens to pinpoint similarities between KMS failure and groupware failure. On the basis of our novel application of Grudin's work in the realm of knowledge work support systems, the paper contributes a set of recommendations for how to integrate KMS with everyday knowledge work. The eight recommendations identified and discussed are: (1) knowledge work support systems must render perceived value as to reinforce user commitment in situations where additional effort is required from knowledge workers interacting with the systems, (2) developers should try to lower systems thresholds by minimizing the amount of extra work required and to build in incentives for use by making salient both individual and collective benefits, (3) knowledge work support systems must acknowledge and coexist alongside existing cultural and social processes, (4) developers of knowledge work support systems should seek solutions with the capability to leverage knowledge workers' day-to-day activities, (5) knowledge work support systems must not be introduced as stand-alone applications but rather as integrated systems exploiting spin-offs from activities knowledge workers are already engaged in, (6) evaluations of knowledge work support systems need to include various knowledge worker groups and be designed as collaborative efforts seeking both individual benefits and organizational effects, (7) well-adapted system support for knowledge work is best achieved through many small applications that cater to specific aspects of knowledge workers' everyday practices, and (8) familiar applications used by a critical mass of employees should be selected as hosts for the knowledge management features to be added in situations where the support systems require input from and interaction with many knowledge workers.

Given our recommendations, we draw on Markus et al.'s (2002) discussion about the poor fit between specific requirements of knowledge work processes and existing IS design theories as to position our knowledge contribution. We are strong in our belief that our recommendations help developers as well as researchers to better understand and overcome the three disconnects as identified by Markus et al. (2002)

First, Markus et al. (2002) argue that today's support systems for knowledge work do not offer the flexibility needed for process emergence. Our recommendations No. 4 and No. 7 provide guidance for how to tackle this disconnect. As knowledge work defies routinization and requires creativity to produce idiosyncratic and esoteric knowledge, such work practice is untidy compared to operational or administrative business processes. Hence, knowledge work support systems must be able to go beyond written instructions and official task descriptions, thus appreciating exceptions not only as something inevitable but also as a necessity. Indeed, the risk of support systems becoming too narrow and rigid can be decreased by exploiting the combined intuition of several different developers.

Second, Markus et al. (2002) assert that existing knowledge work support systems are designed for a known user community, whereas emergent knowledge work practice is characterized by shifting user types having varying knowledge requirements. Our recommendations No. 3 and No. 6 provide guidance for how to tackle this disconnect. As different knowledge worker groups may have various social norms and values, support systems designed for one particular group can have built-in features conflicting with cultural structures of another group. Bearing this is mind, developers can facilitate adoption of systems across groups of knowledge workers by making salient norms and values that underpin their design and reflect upon what intended and unintended consequences these might render. An alternative for KMS developers seeking to promote system adoption in new domains is to become aware of the different set of evaluation criteria for success that may exist in a given context and adjust to these conditions.

Third, analyzing the role of support systems in contemporary knowledge work practices, Markus et al. (2002) argue that knowledge workers either manage their tools instead of attending their work or ignore their tools al-

together. Our recommendations Nos. 1, 2, 5, and 8 provide guidance for how to tackle this disconnect. On a general level, our recommendations reflect the notion that knowledge work support systems must not be isolated and should be integrated into work practice. For the purpose of avoiding situations where knowledge workers manage their systems rather than getting the job done, developers must recognize socio-technical issues associated with disparity in work and benefit. In this way, knowledge work support systems capable of attracting a critical mass of users can be developed. In addition, paying attention to unobtrusive accessibility and the adoption process may deepen developers' understanding of how support systems can be better integrated with both the day-to-day tasks of knowledge workers and their performance of the tasks.

Finally, rather than attempting to draft general laws that must be applied in every situation, in this paper we suggest general recommendations for how to integrate KMS with everyday knowledge work. We suggest that all developers of knowledge work support systems should consider these recommendations in all projects aimed at bridging the knowing-doing gap, although it is clearly up to KMS developers to interpret and apply the recommendations for themselves. We conclude that additional studies focused on how to keep KMS alive — updated, current, maintained — by encouraging use are necessary if IS researchers are to provide useful advice to practitioners on the implementation and use of support systems with the potential to bridge the knowledge application gap in organizations. Indeed, our study offers a wealth of opportunities for further investigations of the many unresolved issues and challenges surrounding deployment of support systems in any intellectual activity involving knowledge application.

REFERENCES

Alavi, M., & Leidner, D. E. (2001). Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly, 25*(1), 107-136.

Blackler, F. (1995). Knowledge, knowledge work and organisations: An overview and interpretation. *Organisation Studies, 16*(6), 1021-1046.

Boland, R. J., & Tenkasi, R. V. (1995). Perspective making and perspective taking in communities of knowing. *Organization Science, 6*(4), 350-372.

Brown, J. S., & Duguid, P. (1991). Organisational learning and communities of practice: Toward a unified view of working, learning and innovation. *Organization Science, 2*(1), 40-57.

Burstein, F. and Linger, R. (2003). Supporting post-Fordist practices — A knowledge management framework for supporting knowledge work. *Information Technology & People, 16*(3), 289-305.

Davenport, T. H., Jarvenpaa, S. L., & Beers, M. C. (1996, Summer). Improving knowledge work processes. *Sloan Management Review*, 53-65.

Davenport, T. H., & Prusak, L. (1998). *Working knowledge*. Boston: Harvard Business School Press.

Garud, R., & Kumaraswamy, A. (2005). Vicious and virtuous circles in the management of knowledge: The case of Infosys Technologies. *MIS Quarterly, 29*(1), 9-33.

Grant, R. M. (1996, Winter). Towards a knowledge-based theory of the firm. *Strategic Management Journal, 17*, 109-122.

Grudin, J. (1987). Social evaluation of the user interface: Who does the work and who gets the benefit? In H. J. Bullinger, & B. Shackel (Eds.), *Proceedings of INTERACT 1987* (pp. 805-811). Amsterdam: Elsevier Science Publishers.

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organisational interfaces. In *Proceedings of CSCW 1988* (pp. 85-93). ACM Press.

Grudin, J. (1994). Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM, 37*(1), 92-105.

Hahn, J., & Subramani, M. R. (2000). A framework of knowledge management systems: issues and challenges for theory and prac-

tice. In *Proceedings of the International Conference on Information Systems 2000* (pp. 302-312).

Hedlund, G. (1994). A model of knowledge management and the N-Form Corporation. *Strategic Management Journal, 15*, 73-90.

Holtshouse, D. (1998). Knowledge research issues. *California Management Review, 40*(3), 277-280.

Kankanhalli, A., Tan, B. C. Y., & Wei, K. K. (2005). Contributing knowledge to electronic knowledge repositories: an empirical investigation. *MIS Quarterly, 29*(1), 113-143.

King, W. R. (1999). Integrating knowledge management into IS strategy. *Information Systems Management, 16*(4), 70-72.

Kock, N., & McQueen, R. J. (1998). Groupware support as a moderator of interdepartmental knowledge communication in process improvement groups: an action research study. *Information Systems Journal, 8*, 183-198.

Lindgren, R., Henfridsson, O., & Schultze, U. Design principles for competence management systems: A synthesis of an action research study. *MIS Quarterly, 28*(3), 435-472.

Lindgren, R., & Stenmark, D. (2002). Designing competence systems: Towards interest-activated technology. *Scandinavian Journal of Information Systems, 14*, 19-35.

Lindgren, R., Stenmark, D., & Ljungberg, J. (2003). Rethinking competence systems for knowledge-based organizations. *European Journal of Information Systems, 12*(1), 18-29.

Mao, J. Y., & Benbasat, I. (1998). Contextualized access to knowledge: Theoretical perspectives and a process-tracing study. *Information Systems Journal, 8*, 217-239.

Markus, L. M., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly, 26*, 179-212.

Miles M. B., & Huberman, A. M. (1984). *Qualitative data analysis. A sourcebook of new methods*. London: Sage Publications.

Nonaka, I. (1994). A dynamic theory of organisational knowledge creation. *Organization Science, 5*, 14-37.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge creating company: How Japanese companies create the dynamics of innovation*. New York: Oxford University Press.

Pfeffer, J., & Sutton, R. I. (2000). *The knowledge-doing gap: how smart companies turn knowledge into action*. Boston: Harvard Business School Press.

Robertson, M., Sørensen, C., & Swan, J. (2001). Survival of the leanest: Intensive knowledge work and groupware adaptation. *Information Technology & People, 14*(4), 334-353.

Schultze, U. (2000). A confessional account of an ethnography about knowledge work. *MIS Quarterly, 24*(1), 3-41.

Schultze, U., & Boland, R. J. (2000). Knowledge management technology and the reproduction of knowledge work practices. *Journal of Strategic Information Systems, 9*, 193-212.

Schultze, U., & Leidner, D. (2002). Studying knowledge management in information systems research: Discourses and theoretical assumptions. *MIS Quarterly, 26*(3), 213-242.

Spender, J. C. (1996). Organisational knowledge, learning and memory: Three concepts in search for theory. *Journal of Organisational Change Management, 9*(1), 63-78.

Stein, E. W., & Zwass, V. (1995). Actualizing organizational memory with information systems. *Information Systems Research, 6*(2), 85-117.

Stenmark, D. (2001). Leveraging tacit organisational knowledge. *Journal of Management Information Systems, 17*(3), 9-24.

Tsoukas, H. (1996). The firm as a distributed knowledge system: A constructivist approach. *Strategic Management Journal, 17*, 11-25.

Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems, 4*, 74-81.

Yin, R. K. (1994). *Case study research. Design and methods* (2nd ed.). Thousands Oaks, CA: Sage Publications.

Dr. Dick Stenmark is a senior lecturer at the department of informatics at Göteborg University, Sweden. His teaching and research interests include knowledge and content management systems and information seeking behaviour amongst non-IR-professionals. Dr. Stenmark has also more than 17 years of experience from industry where he has worked with intranet architecture and design, search engines, and content management systems. His work is published in Scandinavian Journal of Information Systems, European Journal of Information Systems, Knowledge and Process Management, Journal of Management Information Systems, International Journal of Business Environment, *and the* International Journal of Technology and Human Interaction.

Dr. Rikard Lindgren is a member of the management group at the Viktoria Institute, Göteborg, Sweden. He is also an associate professor at the IT University of Göteborg. Dr. Lindgren has published his research in MIS Quarterly, Information and Organization, European Journal of Information Systems, Information Systems Management, Scandinavian Journal of Information Systems, *and other journals in the information systems discipline.*