

Dual information systems for organizational working and learning: the business and breakdown layers

Timo K. Käkölä

Department of Computer Science and Information Systems

University of Turku, DataCity, FIN-20520 Turku, Finland

Tel.: +358 40 5519645, Fax: +358 21 6338600

E-Mail: timo.kakola@utu.fi

Kalle I. Koota

Orion Corporation, Orion-Farmos, R&D

P.O. Box 425, FIN-20101 Turku, Finland

Tel.: +358 21 2727665, Fax: +358 21 2727546

E-Mail: kalle.koota@orion.mailnet.fi

Abstract

The conceptual structure of most computer-based information systems reflects a dualism of technology. During the development phase, part of the work-domain related knowledge is formalized and encoded in the software, making it difficult for users to reflect upon and use this knowledge. This dualism deters the interpretive flexibility of information systems. Dual Information Systems (DIS) are needed that enable and reinforce both effective, institutionalized enactment and questioning and (re)construction of computer-supported work routines. DIS have a three-layered conceptual structure: (1) people draw on the business layer to work and learn; (2) people use the breakdown layer to handle unexpected breakdowns; (3) self-organizing project teams use the project layer to create innovative work and IS (re)designs. We outline the theoretical background, conceptual structure and generic services of DIS. We elaborate on the services and the conceptual design of the business and breakdown layers of DIS (bDIS). The services help people work effectively and develop competence needed to handle breakdowns and participate in the redesign

* Timo Käkölä is responsible for the theoretical development of this paper. Both authors contributed equally to Section 4. Kalle Koota programmed the ReDIS prototype.

project teams. The conceptual design extends Chang's hyperknowledge framework. Finally, we demonstrate the conceptual design and services in a financial services organization with the help of the ReDIS prototype.

Keywords

Act Orientation, Dual Information Systems, Hyperknowledge Framework, Hyperknowledge Organization, Interpretive Flexibility, Knowledge-Creation Nets, Organizational Interfaces, Organizational Creation of Knowledge

1 INTRODUCTION

Information systems researchers have paid considerable attention to developing systems to enhance managerial decision making and learning (e.g., group and executive support systems) and to automate office routines (e.g., workflow systems) (Swanson and Ramiller, 1993). While this research remains relevant, it undermines the need for new knowledge about the development of systems that enable *organizational* creation and sharing of knowledge, not only the learning and decision making of middle managers and executives. Organizations' existing office systems remain at least as important sources of new knowledge as the new coordination technologies. It becomes crucial for researchers to determine how to redesign these systems to facilitate organizational creation and sharing of knowledge.

Redesign presumes changes in the conceptual structure of office systems. But first, one must understand the role of computer-based information systems (CBIS) in organizations and why many systems are inadequate. We draw upon the work of Orlikowski (1992), Nurminen (1988), and Nonaka (1994) in this paper to reach a conceptual understanding. Orlikowski (1992) recognizes that information technology can be seen as enacted by human agency and as institutionalized in structure. She calls this "the duality of technology." Duality implies that organizations can employ information technology as a source of working and learning, if agents* can use and modify the technology whenever it is necessary to redesign computer-supported work practices, and if the technology can be institutionalized as a legitimate component of the organizational working and learning environment.

Unfortunately, too many organizations today suffer from *dualistic, institutionalized* CBIS that hide the constructed nature of CBIS from agents (Käkölä, 1995a). They (1) limit lateral communication, coordination and knowledge sharing; (2) provide little feedback to agents (especially in the lower echelons of an organization) on work arrangements and on the coordination and communication patterns that emerge from their use; (3) limit agents' ability to reflect and inquire within the social and technical contexts in which the agents are embedded, restraining them from creating, questioning, and modifying practical knowledge when problems emerge; and consequently (4) endanger the process of reinvention that any complex technological artifact should undergo when put to use (Ciborra and Lanzara, 1994). For instance, changing the computer-supported work processes through user-driven work and IS design is almost impossible unless agents thoroughly understand the content and organization of work (Hellman, 1989).

* We use the word "agent" to refer to people whose work is computer supported. We want to avoid the term "user" since that term connotes that information systems can be used and studied independently of other types of work.

Dual Information Systems (DIS) are needed that enable both effective, institutionalized working and the questioning and (re)design of computer-supported work. Orlikowski (1992) provides neither a conceptual model nor practical guidelines to help organizations design DIS. This paper continues the efforts of Käkölä (1995a) to do so. Käkölä (1995a) complemented Orlikowski's work by drawing on the act-oriented perspective (Eriksson and Nurminen, 1991). This perspective narrows the dualistic gap by interpreting the functions of computer software as an inseparable component of the work of knowledgeable people (Nurminen, 1988). It helps agents even on the shop floor gain a comprehensive understanding of their work, its computer-supported parts, and its relationship to the business as a whole. In light of the perspective, a necessary but insufficient condition for a CBIS to qualify as a DIS is that it helps agents develop such an understanding.

In Section 2, we outline the theoretical background of DIS. In Section 3, we outline the conceptual structure of DIS and elaborate on the services of the business and breakdown layers of DIS (bDIS). The services help people work effectively and develop competence needed to handle breakdowns and redesign work. Next, we present and extend the hyperknowledge framework (Chang, Holsapple, and Whinston, 1994) to develop the conceptual design of bDIS. We conclude Section 3 by discussing the use of work and software process modeling in Dual Information Systems. In Section 4, we demonstrate the conceptual design and services in a financial services organization, describe their potential organizational benefits, and encourage commercial implementations with the help of the ReDIS prototype. We state our conclusions and discuss issues for future research in the last section.

2 DUALITY OF TECHNOLOGY, ACT ORIENTATION, AND HYPERTEXT ORGANIZATION

To design Dual Information Systems, a comprehensive theoretical understanding of the role of CBIS in organizational working, the creation of knowledge, and learning is necessary. We recapitulate the work of Orlikowski (1992), Nurminen (1988), and Nonaka (1994) in this section to provide the requisite background.

2.1 Duality of technology, interpretive flexibility, and time-space disjuncture

Orlikowski (1992) introduces three constructs that are central to this paper: the duality of technology, its interpretive flexibility, and time-space disjuncture. The duality construct sees information technology as enacted by human agency and as institutionalized in structure. Designers produce a technology to provide resources and rules by creating and encoding work-domain related knowledge into it. Agents socially construct a technology by assigning it different meanings and using it flexibly in their work. But technologies usually become reified and institutionalized over time because agents cannot continuously reinterpret or physically modify them, if the agents are to accomplish their work efficiently.

The duality of technology recognizes that "technologies are products of their time and institutional context, and will reflect the knowledge, materials, interests, and conditions at a given locus in history" (Orlikowski, 1992, p. 421). However, the dual nature of CBIS is masked by the time-space disjuncture arising from the various phases (e.g., design, implementation, and employment) of interaction between a technology and organizations. Within and between these phases, the actions

constituting the technology are separated temporally and spatially from the actions constituted by the technology.

The interpretive flexibility construct aims to collapse the time-space disjuncture. It emphasizes that “there is flexibility in how people design, interpret, and use technology, but that this flexibility is a function of the material components comprising the artifact, the institutional context in which a technology is developed and used, and the power, knowledge, and interests of human actors . . .” as well as time (Orlikowski, 1992, p. 421). Orlikowski (1992, p. 421) summarizes the causality between the time-space disjuncture and the interpretive flexibility of technology as follows: “The greater the temporal and spatial distance between the construction of a technology and its application, the greater the likelihood that the technology will be interpreted and used with little flexibility.”

2.2 The dualism of information technology: implications and alleviations

The interpretive flexibility of information technology is often poor (Bødker and Grønbaek, 1991; Eriksson, Hellman, and Nurminen, 1988; Gasser, 1986; Tyre and Orlikowski, 1994; Zuboff, 1988). This inflexibility results in part from agents with insufficient shared knowledge of the nature of social practices as a whole; of the articulation of these practices in time and space by the structural properties of organizations; of their own roles in the organization; and of the role of information systems as a structural property mediating work processes.

One important reason for agents' lack of awareness is that the conceptual and material structures of computer software reflect a *dualism* of technology; the constructed nature of CBIS is masked by the software. Information systems separate symbolic information from the material and social systems the symbols represent, hide the processing rules and retention structures in the software and database schemas, and blur the role of people as the producers and consumers of information (Boland, 1991; Nurminen, 1988).

The prevalent dualistic structure and poor interpretive flexibility of information technology have costly implications. Agents are restricted to using functions expressed in the software (Kogut and Zander, 1992, p. 390). They also face considerable difficulties monitoring their actions since they cannot fully interpret and validate the meaning of information produced by the systems, and they cannot see and feel the outcomes of their computer-supported actions (Zuboff, 1988, pp. 79-96). Because of their limited ability to control all aspects of work, including computerized tasks, the agents cannot necessarily be responsible for their work as a whole. Finally, the agents' ability to intervene in and transform existing social practices is limited because the agents cannot easily criticize and challenge the interpretative schemes, resources and norms embedded in the algorithms and databases of the CBIS (Lyytinen and Ngwenyama, 1991; Orlikowski, 1992).

Agents can regain control of their jobs in computer-supported work environments. Orlikowski (1992, p. 418) states: “. . . knowledgeable and reflexive human agents are capable of altering the controlling influence of the technology. The extent to which individuals modify their use of technology, however, depends on whether they acknowledge its constructed nature. This is determined by the degree to which individuals can recognize the mediating role of technology, can conceive of an alternative beyond it, and are motivated to action.” Thus, agents' ability to control their work depends on their domain and technology-related skills and knowledge as well as the shared stocks of knowledge that inform them about their own and other agents' roles and normatively regulated and sanctioned behaviours (Lyytinen and Ngwenyama, 1991; Zuboff, 1988).

The conceptual structure and services of DIS must help agents control and redesign their work as a whole, including its computerized aspects. For this purpose, DIS need to help bridge the time-space

disjuncture both vertically (between users and designers) and horizontally (between agents coordinated through the systems).

2.3 Act-oriented perspective: toward DIS on the level of human agency

The time-space disjuncture between the design and use of CBIS and the dualistic nature of computer software imply that agents often cannot see the link between their work as a whole and its computerized parts. Orlikowski (1992) provides no conceptual solution to uncover this missing link.

Käkölä (1995a) used the act-oriented perspective (Eriksson and Nurminen, 1991; Nurminen, 1988) to provide a solution. According to this perspective, a CBIS cannot be separated from agents' work because no CBIS can serve as a conscious actor. Therefore, every computerized task must have a responsible human agent. Conceptually, each agent has his or her own information system receiving, memorizing, processing, and transmitting the information necessary for his or her work.

The act-oriented perspective bridges the time-space disjuncture *vertically* by seeing the knowledge encoded in software by designers in time-space context A as the acts of the responsible agents in time-space context B and *horizontally* by making explicit the coordinating role of information systems, even when the computer-supported acts of agents are inter-connected through shared objects of work in integrated databases. Once this bridge has been built, agents can better exploit the interpretive flexibility of Dual Information Systems to work and share knowledge.

2.4 Hypertext organization: toward DIS on the organizational level

Knowledge-creating organizations are required to institutionalize reflection-in-action in their processes and structures so that their members need not depend on the established wisdom (Nonaka, 1994). Analogously, they require Dual Information Systems that support institutionalized reflection-in-action. Nonaka (1994) proposes 'hypertext organization' as an organizational design prototype for institutionalized reflection-in-action. Hypertext organization is a *dual* organizational structure: it "coordinates the allocation of time, space, and resource within the organization" (Nonaka, 1994, p. 33) so that an organization can achieve high performance in routines and simultaneously ensure long-term survival by creating and applying new knowledge. Therefore, the conceptual structure of Dual Information Systems must reflect the hypertext organization structure.

Hypertext organization is formed by the dynamic combination of hierarchically organized business units and self-organizing project teams that pursue the equivocal visions of the top management by drawing upon and accruing organizational knowledge base. Nonaka (1994, pp. 32-33) states: "The core feature of the hypertext organization is the ability to switch between the various 'contexts' of knowledge creation to accommodate changing requirements from situations both inside and outside the organization. . . . Each context has a distinctive way of organizing its knowledge creation activities. . . . Hypertext organization design first distinguishes the normal routine operation conducted by a hierarchical formal organization from the knowledge creating activities carried out by self-organizing teams. . . . By establishing the most appropriate organizational setting for the two activities, an organization can maximize the efficiency of its routine operation, which is determined by bureaucratic principles of division of labor and specialization, and also the effectiveness of its knowledge creation activities."

Hypertext organizations are comprised of three layers; knowledge-base, business-system, and project-system (Figure 1). The 'knowledge-base' layer "embraces tacit knowledge, associated with organizational culture and procedures, as well as explicit knowledge in the form of documents, . . . computerized databases, etc" (Nonaka, 1994, p. 33). Normal work routines are enacted by a formal,

bureaucratic organization in the ‘business-system’ layer (hereafter “business layer”). The ‘project-system’ layer (hereafter “project layer”) provides a field of interaction where loosely linked project teams create knowledge.

In a hypertext organization, knowledge is created through the circular movement of agents among the three layers. Members of project teams are selected from different functions and business units across the business layer. They interact with the knowledge-base layer at the bottom and make an ‘inventory’ of the knowledge acquired and created in the project layer. “After categorizing, documenting, and indexing the new knowledge, they come back to . . . business-system layer and engage in routine operation until they are called again for another project” (Nonaka, 1994, p. 33).

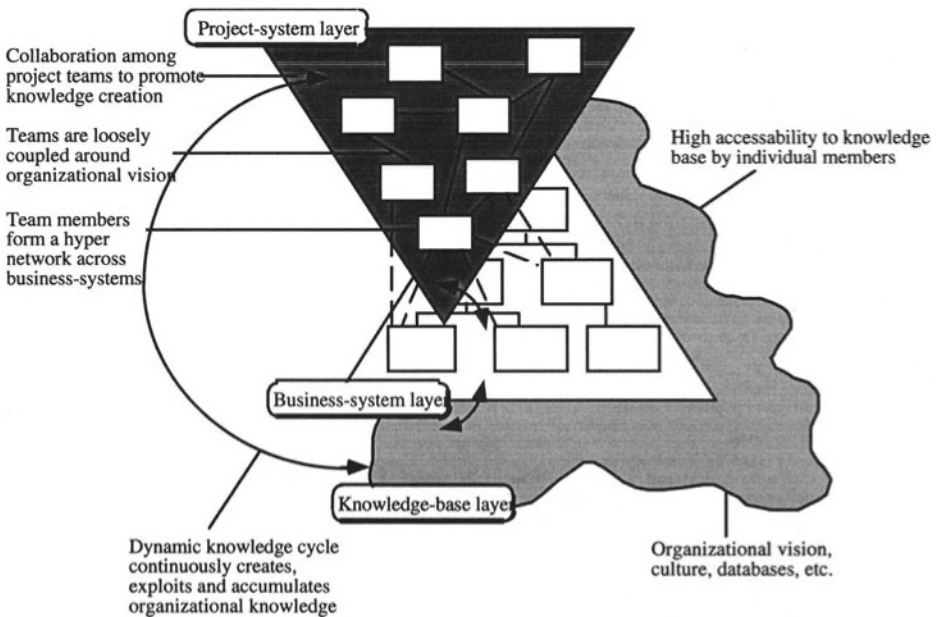


Figure 1 Hypertext Organization – An Interactive Model of Hierarchy and Nonhierarchy (Nonaka, 1994, p. 34).

3 DUAL INFORMATION SYSTEMS AND COMPUTER SUPPORTED WORK

3.1 The conceptual structure of DIS

Dual Information Systems (DIS) enable and reinforce both effective, institutionalized enactment and questioning and (re)construction of computer-supported work routines. In accordance with our theoretical framework, they have a three-layered conceptual structure (Figure 2): (1) agents continuously draw on the business layer of DIS to work, learn, and coordinate routines in the

business units; (2) agents use the breakdown layer of DIS to zoom in on the details of their work to handle unexpected (coordination) breakdowns; (3) self-organizing project teams use the project layer of DIS to create innovative work and IS (re)designs that can be enacted on the business layer. DIS are implemented by combining a coordination technology platform (Rein, 1992) with hyperknowledge systems (Chang, Holsapple, and Whinston, 1994). Consequently, we call hypertext organizations that enable and are enabled by DIS hyperknowledge organizations.

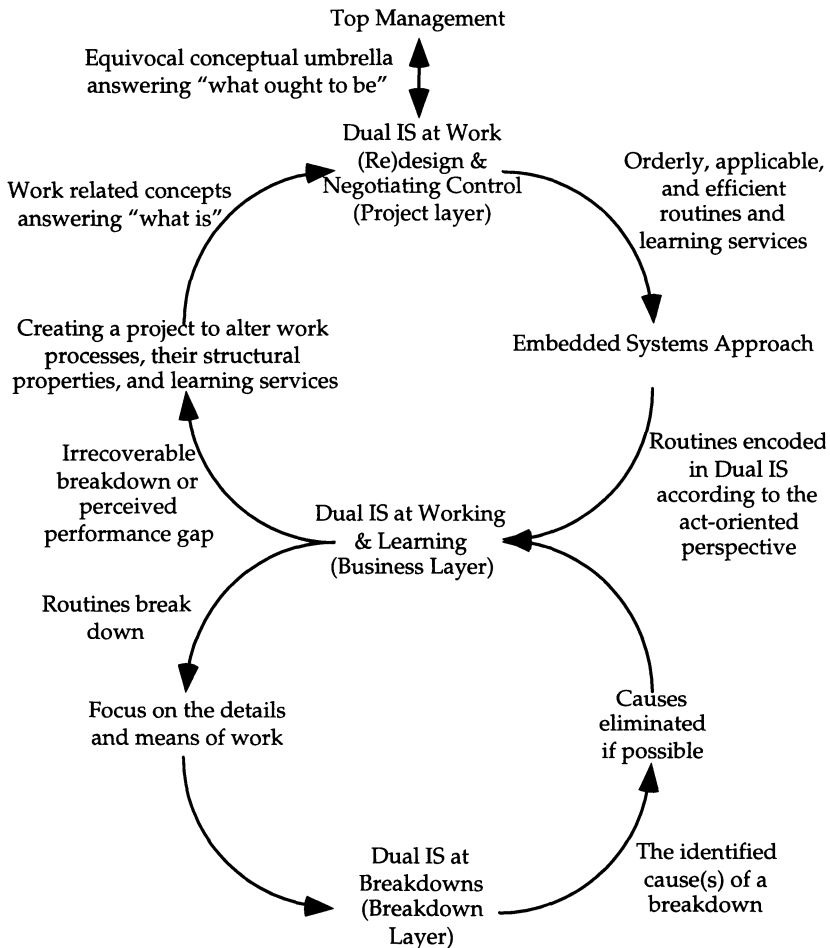


Figure 2 The conceptual structure of Dual Information Systems.

From the point of view of institutionalized reflection-in-action, the importance of routine use of information systems cannot be over-stated. Information systems as a structural property of organizations remain constitutive of organizational realities that can be known, understood, and acted upon (Lyytinen and Ngwenyama, 1991; Orlikowski, 1992; Orlikowski and Robey, 1991; Tyre

and Orlikowski, 1994). For example, Tyre and Orlikowski presented and analyzed three case studies and found (1994, p. 111) that “. . . routine use was . . . necessary for on-going adaptation; it provided the raw data that, if utilized, could lead to improvements in the technology or the way it was applied in the local context.” Furthermore, the creation of new knowledge to improve work in ambiguous circumstances is rarely possible without the objective information obtained by monitoring and measuring organizational routines, because this information can be used to refute or support and legitimate more subjective interpretations (Daft, Lengel, and Trevino, 1987). For these reasons, we focus on the business and breakdown layers of DIS (bDIS). Käkölä and Koota (1996) discuss the project layer.

3.2 Business and breakdown layers of DIS

Käkölä (1995a) presented the Embedded Systems Approach (ESA) to instantiate the business and breakdown layers of DIS into Embedded Application Systems (Eriksson and Nurminen, 1991). Embedded Application Systems follow the act-oriented structure (Eriksson and Nurminen, 1991). They enable the effective enactment of routines in the business units and allow agents to reflect and inquire into their work practices as a whole, including computerized tasks. In a continuous knowledge-creating spiral, ESA relies on and enables the project teams of the hyperknowledge organization. The teams develop new concepts of computer-supported work and share them among the business units. Work processes are externalized, visualized, and memorized by using organizational and software process modeling formalisms. Process models (subsection 3.5) serve as the building block of the organizational interface (Malone, 1985) through which the computerized tasks and knowledge become inseparable components of the agents' working and learning environment. Breakdown management services let agents zoom in on the details of their work practices, check shared databases for mistakes, and fix many breakdowns locally. Additional learning services can easily be provided. We will elaborate in subsection 3.4 and section 4.

3.3 Hyperknowledge as a basis of Dual Information Systems

Dualistic technologies embed work-related knowledge in the form of linearly organized, fairly immutable concepts and procedures. These concepts and procedures are hidden in the software and tend to promote the institutionalization of work practices to those envisioned by designers in earlier time-space contexts. Consequently, DIS must transcend these linear structures and let agents organize computerized information, concepts, and work procedures as flexibly as they organize their manual work practices and materials. In this way, agents can work effectively and construct multiple interpretations of work in a hyperknowledge organization.

Hyperknowledge (Chang, Holsapple and Whinston, 1994) is a promising way to design DIS with high interpretive flexibility. It is strongly related to the concepts of hypertext and hypermedia (Bush, 1945; Engelbart, 1963), and to the architecture of hypertext systems (Nelson, 1987; Nielsen, 1990). However, the idea of hyperknowledge is wider than those of hypertext or hypermedia. Hyperknowledge is an ideal working and learning environment that holds work-related knowledge and, at the same time, defines the nature of hypertext or hypermedia. Such an environment partly eliminates the reifying effects of the time-space disjuncture: the designers still encode a part of work-related knowledge into the environment, but now this knowledge is transparent and partly nonlinear. Agents can navigate flexibly in the environment, widen their knowledge, and actively externalize their tacit knowledge and combine new knowledge into the environment.

Chang, Holsapple, and Whinston (1989; 1993; 1994) originated the framework of the hyperknowledge environment. It recognizes that an agent "cognitively possesses many diverse and interrelated pieces of knowledge (i.e., concepts). Some are descriptive, others are procedural in nature, yet others are concerned with reasoning, and so forth. The mind is able to deal with these concepts in a fluid and inclusive manner via controlled focusing of attention. In effect, the decision maker actively acquires (i.e., recalls, focuses on) desired pieces of knowledge by cognitively navigating among the universe of available concepts" (Chang, Holsapple, and Whinston, 1993, p. 30). Dual Information Systems should be designed in the same manner so that they become natural extensions of people's working and learning environments.

According to the hyperknowledge framework (Figure 3), the configuration of a hyperknowledge environment consists of three components: the Language and the Presentation System, the Problem Processing System, and the Knowledge System. The Language and the Presentation System mediate messages to and from the hyperknowledge environment. The Problem Processing System contains the dynamic part of the system and may possess inferential abilities. The system consists of three different parts – the data manager, model manager and dialogue manager – with which it can handle all the user requests or responses to and from the various knowledge sources in the system. The Problem Processing System is capable of presenting the knowledge in different ways, as chosen by the user through the user interface. The Knowledge System contains all the encoded, explicit knowledge, which can be descriptive, procedural, linguistic, reasoning, etc. It stores, in groups, concepts that are related to each other by definition and/or by association.

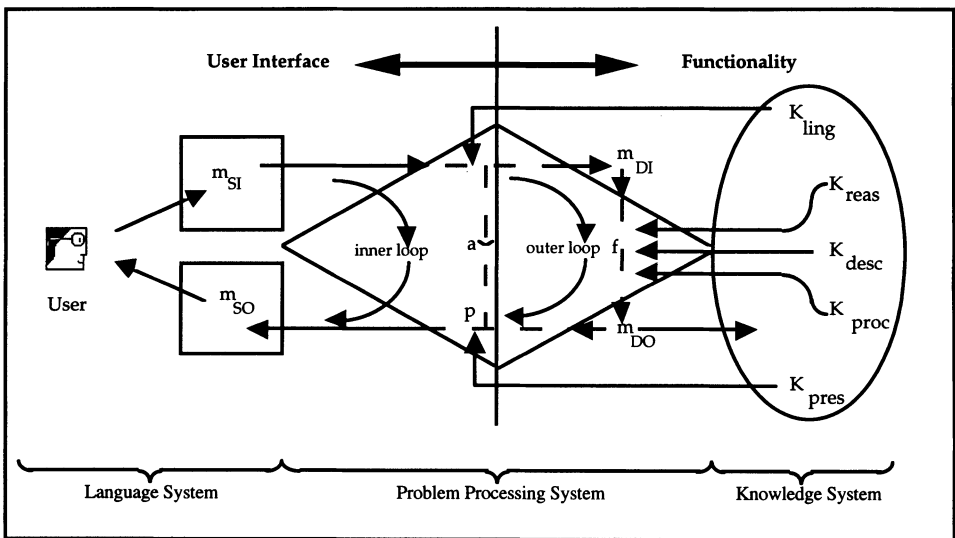


Figure 3 The hyperknowledge framework (Chang, Holsapple, and Whinston, 1994, p. 482).

Figure 3 depicts the hyperknowledge framework. The symbols in Figure 3 signify the following:
 m_{SI} = surface input message; key strokes, mouse movement, touching the screen, etc.

- mDI = deep input message; specifies what the Problem Processing System must do in order to satisfy the request expressed in mSI
- a = an assistance function; a request for further elaboration, a correction or error message
- f = a function which combines new knowledge from the available explicit knowledge
- p = a presentation function
- K_{ling} = linguistic knowledge available to the Problem Processing System in the Knowledge System
- K_{reas} = reasoning knowledge available in the Knowledge System
- K_{desc} = descriptive knowledge available in the Knowledge System
- K_{proc} = procedural knowledge available in the Knowledge System
- K_{pres} = presentation knowledge available in the Knowledge System
- mDO = deep output message; derived knowledge produced by the Problem Processing System
- mSO = surface output message to the user.

In an ideal hyperknowledge environment, the agent first receives an impression, possibly several impressions, of the same underlying concept. Each impression presents an image or images (visualization) on the user interface screen. The negative is an internal characterization of the image created by the problem processor. The agent sees the concept through several windows and can thus internalize a holistic view of the concept. To avoid the problems presented by Conklin (1987), the “lost in space” and “cognitive overhead” feelings, the hyperknowledge environment needs concept maps to guide both the designers and the agents (Chang, Holsapple, and Whinston, 1994).

Hyperknowledge framework is somewhat limited to the working, learning, and decision-making of individuals, yielding systems that primarily augment the intellect of individuals. Organizations may not become significantly more efficient or effective simply by giving everybody a definitive and associative access to organizational knowledge. Uncontrolled, unguided exploration and extension of available knowledge are unlikely to converge into meaningful concepts (Nonaka, 1994).

3.4 The conceptual design of the business and breakdown layers of DIS

To remedy the individualistic orientation of the hyperknowledge framework, we develop the conceptual design of the business and breakdown layers of DIS in this section. The Embedded Systems Approach uses the *organizational role* concept (Roos and Starke, 1981; Stryker and Statham, 1985) to divide each business unit in the business layer of a hyperknowledge organization into a set of functional work groups. Each group is responsible for a clearly defined part of the work process(es) of a business unit. Each agent in a functional group enacts the same work role.

The efficiency and effectiveness of the approach depend on two factors. First, agents must be able and motivated to internalize specialized role-centric language and norms and to exploit the role-centric information resources to a maximum degree. Second, agents must be offered redundant information to help them construct shared stocks of knowledge about the interconnections between their computer-supported work routines and the organizational work and performance. Redundant information helps agents recognize their location in an organization, interact, develop a common direction, create new concepts, enter each others' spheres of operation, provide advice, fix coordination breakdowns, and even do each other's jobs if necessary (Nonaka, 1994).

The factors promote both efficiency and effectiveness. When clear role-specific knowledge and resource requirements have been established, organizations can offer learning services and agents can use them to internalize quickly the skills necessary for role enactment. A shared language helps agents share tacit knowledge and use contradictions in their perspectives as a source of new knowledge to improve work practices (Nonaka, 1994; Zuboff, 1988).

Clear boundaries between shared and role-centric stocks of knowledge are often difficult to determine. Moreover, the organizational creation and sharing of knowledge changes these boundaries over time: roles are merged or the job content and knowledge requirements are upgraded in other ways. The distinction is made for analytical and design purposes only.

The organizational interface of bDIS must offer agents a Language System with a balanced mix of shared and role-centric technology-enabled and constrained languages (Figure 4). To comply with the act-oriented perspective, the role-centric language defines institutionalized, prenegotiated responsibility for and control of the means and objects of work. For this purpose, bDIS must provide at least the following role-centric services: (1) *Work enactment and coordination* services help agents allocate resources and take care of their routines. (2) *Learning* services help agents develop a theoretical understanding of work, including its computerized aspects, needed to enact roles effectively. (3) *Performance monitoring* services help agents monitor both their work activity and its results. They also help agents benchmark their performance with agents within functional group(s) (Spendolini, 1992). They promote internal competition and motivate agents to use the other services of bDIS effectively and to figure out ways how to exceed performance expectations; all these are of great importance to accelerate the creation of knowledge (Nonaka, 1994).

No matter how well people do their work, routines may break down unexpectedly. Agents face two challenges in often chaotic breakdown situations. First, they must get their routines back on track as quickly as possible. Second, they must develop and test multiple hypotheses about what went wrong to ensure a lasting solution (Zuboff, 1988). In this respect, breakdowns play a fundamental role in stimulating the creation of knowledge (Heidegger, 1977; Nonaka, 1994; Winograd and Flores, 1986).

Breakdown management services help agents meet both challenges. When routines break down, they let agents shift their focus of attention from normal objects of work to the routinized patterns of interaction and the means of work, including the rules and resources afforded by software (Käkölä, 1995a). Hyperknowledge is especially useful here because the definitive and associative relationships between different chunks of knowledge in the Knowledge System help agents shift focus rapidly. For example, agents can perform audit trails to trace what happened earlier to artifacts (e.g., documents) for which they are responsible and use the learning services to understand what should have happened. If they hypothesize that the problem stems from mistakes of agents in other roles, they can use the redundant information services to better understand how and why such mistakes are possible.

If the analysis reveals that the breakdown stems from coordination or other organizational problems spanning multiple functional work groups, agents can discuss the problems with middle managers. A project team can then be assembled to reconstruct computer-supported work practices and learning services to help eliminate in advance similar breakdowns. The project layer of DIS supports work redesign (Käkölä and Koota, 1996).

A shared language is developed by sharing redundant information according to the norms and rules negotiated and agreed upon in the project layer of a hyperknowledge organization. For this purpose, the business layer of DIS offers agents in functional work groups at least the following redundant information services: browsing, simulated enactment, and benchmarking of inter-connected roles (i.e., the roles interacting with an agent directly or indirectly through Embedded Application Systems). In this way, agents can experiment with the jobs of their colleagues in inter-connected

functional work groups and understand bottlenecks in work performance. The norms and rules preserve the autonomy and privacy of agents by defining what information is shared and what is private. They can be reinforced by encoding some of them in the services of bDIS (e.g., through role-centric access rights to the Knowledge System of bDIS).

We divide the Knowledge System of bDIS into role-centric and shared components (Figure 4). The role-centric component is further divided into two subcomponents: tasks that people do and computerized tasks that Embedded Application Systems do for the people. Role-Connected Task Knowledge externalizes how people should do their routines (enactment knowledge), what are the objects of work (retention knowledge such as paper documents), what are the performance and skill requirements of successful enactment, and where work is physically done (ecological knowledge). These concepts are intertwined and enable the role-centric services of bDIS. For example, performance requirements must be tied explicitly with enactment knowledge so that people can use benchmarking to find and eliminate weaknesses in work performance. Role-Connected Computerized Task Knowledge presents a role-centric view of Embedded Application Systems. In accordance with the act-oriented perspective, the view defines the computerized aspects of work and relates them explicitly to the work of individuals in functional groups.

The shared Knowledge System is also divided into two parts: (1) Inter-Connected Role Knowledge views the work process of a business unit as a set of temporally and spatially distributed interacting roles. It also includes generic knowledge of the objects of the process (e.g., products and services), external stakeholders (e.g., suppliers and customers), and the goals that connect these concepts to a whole. The staff knowledge component stores knowledge of performance, skill profiles, and availability of agents. It supports resource allocation and coordination of work. (2) Inter-Connected Computerized Role Knowledge presents an inter-connected role view of Embedded Application Systems. The view defines how the work of agents in different functional groups is coordinated and mediated by the systems.

The Problem Processing System dovetails with the Language System on the organizational interface side, and with the Knowledge System on the functionality side. The term "Problem Processing System" in the conceptual design of bDIS emphasizes that 'routine work' involves significant problem-solving and articulation work and must be supported by information systems to the same extent as 'decision-making work' (Gasser, 1986; Gerson and Star, 1986; Wynn, 1991).

3.5 Process modeling and dual information systems

How can organizational interfaces be developed to enable effective working, the creation of knowledge, and learning through smooth navigation in DIS? Most traditional structured systems analysis and design approaches are ill-suited to the development of organizational interfaces because (1) they focus almost exclusively on modeling data flows, data structures and other components of *technical* systems and (2) the models are difficult for agents to understand and use (Curtis, Kellner, and Over, 1992). These modeling languages reinforce the development of dualistic, reified information systems (Käkölä, 1995a; El Sawy and Khorshid, 1994, p. 12).

Fortunately, process modeling languages (Curtis, Kellner, and Over, 1992) remedy many of the weaknesses of the technically oriented modeling languages. Curtis, Kellner, and Over state (1992, p. 75): "Process modeling is distinguished from other types of modeling in computer science because many of the phenomena being modeled must be enacted by a human rather than a machine." Käkölä (1995a; 1996) found the Role Interaction Net process modeling language (Rein, 1992) is reasonably effective in alerting people to the dual nature of information systems. Drawing on this research, we

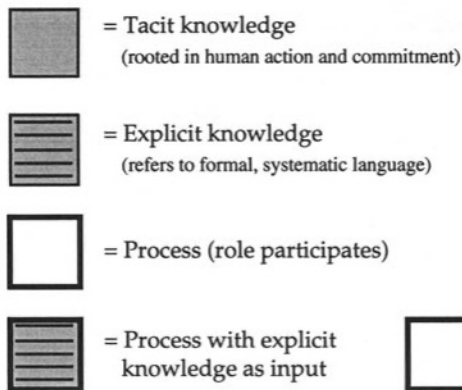
have developed the Knowledge Creation Net (KCN) language (Koota, 1995; Koota and Käkölä, 1996).

The KCN language is based on organizational role (Roos and Starke, 1981; Stryker and Statham, 1985) and knowledge creation theories (Nonaka, 1994), Petri nets (Peterson, 1981), and Role Interaction Nets (Rein, 1992). In accordance with role theory, the language provides two primitives: roles and interactions. A KCN is composed of a set of concurrent roles. The behaviour of a role is described by its solitary actions and its interactions with other roles. Unlike the RIN language, the KCN language also describes how knowledge is created, shared, and converted in interactions between agents. This helps clarify the dual nature of information systems. People create and share tacit and explicit knowledge through the processes of socialization, externalization, combination, and internalization (Nonaka, 1994). Computerized systems enable and constrain these processes by combining existing knowledge according to the rules and procedures encoded into these systems by people in earlier time-space contexts. This distinction between what people do and what computers 'do' can be made explicit when the Knowledge Creation Nets are employed in DIS. Figure 5 clarifies a few notations of the KCN language.

Techniques from Petri nets are used to give the language process description and enactment capability. Enactment capability means that the KCN models can be directly executed on a proper coordination technology platform (Rein, 1992). This is vital for DIS. First, contrary to static models, executable models provide organizations with clear incentives to maintain them. Second, executable models can be used both to measure the performance of work processes and to capture deviations between the specifications and the way work is really done. Consequently, the variances in performance and the underlying reasons for these variances can be quickly identified and fixed, and specifications can be updated to reflect improved work practices.

We illustrate the use of Knowledge Creation Nets in DIS with the help of the ReDIS prototype.

Basic Elements



Knowledge Creation

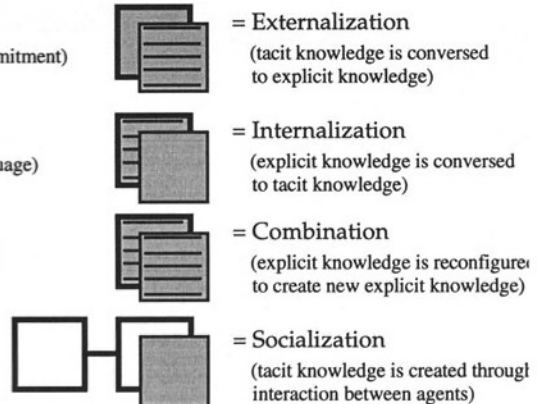


Figure 5 Notations of the KCN language (Koota, 1995; Koota and Käkölä, 1996).

4 WORKING AND LEARNING WITH THE ReDIS PROTOTYPE

In this section, we present the ReDIS* prototype to illustrate the business and breakdown layers of Dual Information Systems. Käkölä and Koota (1996) describe the project layer of ReDIS. The case organization and the work processes described in subsection 4.1 are fictitious, but the ideas stem partly from the IBM Credit Corporation case (Hammer and Champy, 1993). The case study gives ample opportunity to research the development and use of hyperknowledge-based DIS to support organizational coordination and the creation and sharing of knowledge. The ReDIS prototype helps business units process credit requests and issue credit. The ReDIS project was conducted in a laboratory environment. Personal computers and the Microsoft Access™ system (Jennings, 1993) were selected as the development platform.

4.1 The case organization and its credit issuance process

We make the following assumptions about the case organization and the role of ReDIS in the organization. (1) The Credit Corporation is a subsidiary of a much larger organization. (2) Issuing credit effectively is important for the organization as a whole because financing customers' purchases can be extremely profitable. (3) The Credit Corporation is organized as a hyperknowledge organization. The business units that issue credit within their regions form its business layer. (4) ReDIS systems support each business unit locally.

We focus on one business unit. Its credit issuance procedure goes as follows (Figure 6). After negotiating a sale, a sales agent in the Sales unit writes a standardised credit request and sends it electronically to the Credit unit for his region. The creditworthiness of the customer is checked. Rejected requests are sent back to the sales agent. Approved, standard loan contracts are modified to meet customer requirements. An interest rate is determined. A quote is developed and sent back to the Sales unit.

Workers in different functional roles are responsible for their own work. If they cannot accomplish their tasks, they can seek help from "specialists," who are typically employees with more experience and expertise. Relying on a specialist slows down the procedure but is sometimes unavoidable. If workers face a breakdown (e.g., an error in a credit request), they can enter into others' spheres of operation and perform audit trails to see what has gone wrong. Spontaneous quality assurance in every step of the process is an integral part of the organizational culture of the Credit Corporation.

Different characteristics of a work process such as time, quality, and expense are measurable by analyzing various key figures. Each process step is measured by analyzing:

- _ Number of people involved in the step.
- _ Average processing time of credit requests.
- _ Average queuing time for the step.
- _ Percentage of requests that require a specialist's help.

Agents use these measurements to benchmark their performance with agents in the same and inter-connected roles. Work redesign project teams use them and many other measurements to continuously improve the credit process and to periodically redesign it (Käkölä and Koota, 1996).

* The ReDIS project (1994-1996) is led by Timo Käkölä at University of Turku.

4.2 The main components of the business and breakdown layers of ReDIS

Organizational interface

The organizational process models constructed using the Knowledge Creation Net (KCN) language serve as the organizational interfaces of the business units in ReDIS. The organizational interface of each unit has four basic functions. First, it provides agents with role-centric views of the Embedded Credit Application System (ECAS). These views let agents see both the manual and computerized parts of their work as a whole, and thus carry out their responsibilities. Second, agents can use the organizational interface to enact the role-centric functionality of ECAS. Third, it provides a full-fledged hyperknowledge environment with the services envisioned in the conceptual design of the business and breakdown layers (section 3.4); agents can navigate, run different types of queries, and internalize redundant, real-time information about the business unit, its work processes and performance, and ECAS. Fourth, it provides access to the project layer of ReDIS (Käkölä and Koota, 1996).

The performance monitoring services of ReDIS collect performance data from each process enactment and store them in the Knowledge System.

The knowledge system

The procedural and descriptive chunks of knowledge visualized by the KCN model are stored in the Knowledge System of ReDIS. The system is implemented as a relational data base that holds all relevant information about the business unit, the cast (agents associated with certain roles), the process, and process enactment (the number and characteristics of agents in different roles, and the interactions that occur during the process). Naturally, all the relational tables accessed by ECAS are part of this larger knowledge system.

The problem processing system

An interactive working and learning environment is generated by the Problem Processing System that is driven by the Knowledge System's procedural knowledge in the form of Access™ scripts and KCN models.

4.3 Using the business and breakdown layers of ReDIS

We use working scenarios to illustrate how ReDIS fosters the enactment, coordination, and learning of computer-supported work processes. We assume that our imaginary persons, a credit checker and a clerk, have been trained for their jobs, but that they have not yet internalized the process of issuing credits. They do not work in other roles.

The credit checker begins to work by entering ReDIS with his user id and password. The system now knows that he is working at the workstation, registers him as "present" in the Persons table of the staff knowledge system, and determines his role-connected rights and responsibilities. The KCN model of the credit issuance process appears on the screen. It serves as a concept map of the organizational interface to help agents construct a shared understanding of the process (Figure 6). The credit checker can see the different roles and tasks involved, and realize that he is an integral part of the unit with an important organizational function, not just another part of a mechanistic system.

The process fragments within the rounded rectangle visualize the inter-connected role view of the Embedded Credit Application System. In the earlier research of the first author, each computerized task performed by an Embedded Inventory Application System was visually represented in the role

column of the agents responsible for the task. This design reflected the act-oriented perspective: the inseparability of agents' manual and computerized tasks. However, when this design was tested in a laboratory environment, users sometimes mixed up manual and computerized tasks (Käkölä, 1996). Therefore, we have visualized the computerized tasks under the role ECAS. The database column facilitates the bridging of time-space disjuncture horizontally by visualizing the objects of work (here credit request documents) that mediate interactions between agents in different roles. Consequently, the structuring of work in time and space by ECAS is made explicit: agents can see when, where, how, and by whom these objects are stored and retrieved.

The number of requests in different stages of the work process is shown by the organizational interface in real time. This up-to-date reflection of work brings the KCN model to life and gives agents a sense of control over their work. Twenty-two requests are now waiting for the credit check.

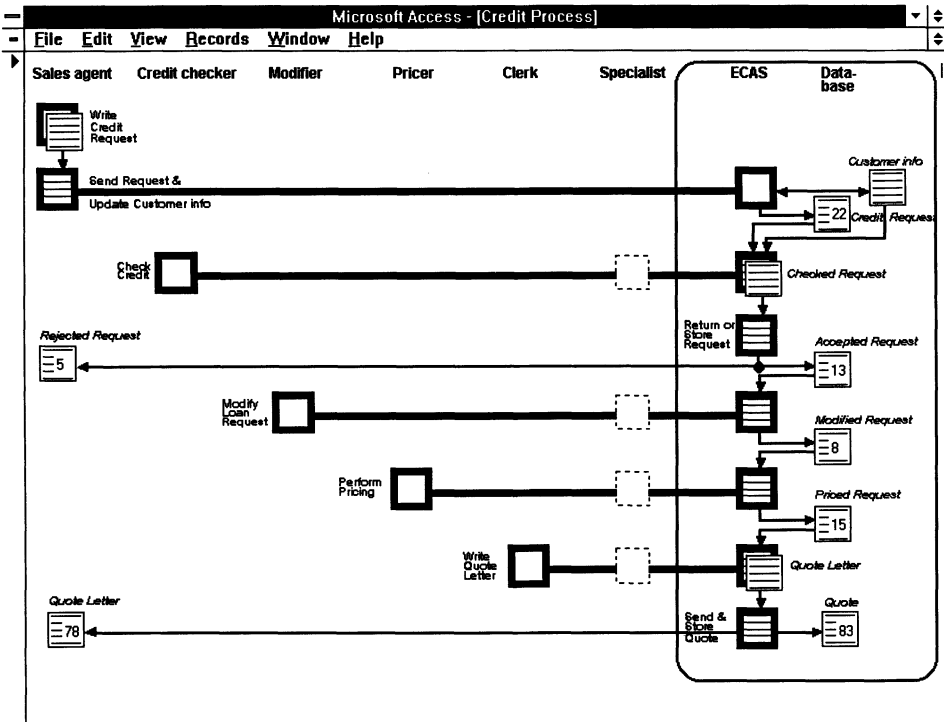


Figure 6 The organizational interface of the business unit.

The organizational interface provides agents with more detailed knowledge of the business unit and its processes to help coordinate and control work effectively and to handle breakdowns. The credit checker can use the KCN-based interface to zoom in on the details of the work processes. In Figure 7, he has opened the Process Description form to view the interactions and ECAS-modules of the Check Credit task. The KCN model shows that credit checkers use ECAS to retrieve queuing credit requests from the data base, and then approve or reject the requests, possibly with the help of

specialists. ECAS automatically provides customer information, stores approved requests in the data base and sends rejected requests back to sales agents.

Learning services related to each interaction are available by clicking the "Instructions" button. The credit checker can click the KCN-icons of the ECAS modules to see the role-centric computerized tasks and the resources and rules designers or colleagues have encoded in these tasks. He can tailor these rules according to the norms the work redesign project teams have negotiated using the project layer of DIS. These services help bridge the time-space disjuncture vertically between designers and workers.

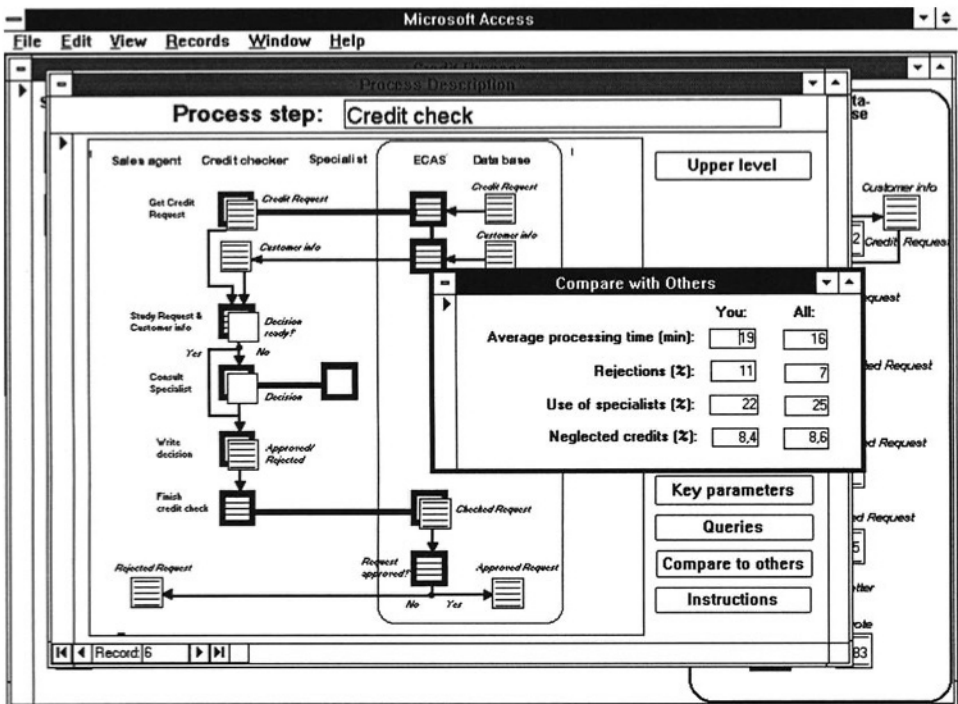


Figure 7 The organizational interface of the credit checkers. The role-centric benchmarking service has been activated.

The worker can run predefined queries, make own queries, and associatively navigate in the Knowledge System to retrieve information about different roles, interactions, and people. It is also possible to do role-centric benchmarking. The credit checker clicks the "Compare with others" button of the Process Description form (Figure 7) and enters the three-month time frame as a parameter to compare his performance over the last three months to the average of the credit checkers in his business unit. ReDIS automatically executes all the predefined queries of the "credit check" process step. It selects from the Interaction table, which stores process enactment and resource use information for performance monitoring and breakdown management services, the credit-check interactions of the business unit during the last three months and uses their starting and

finishing times to calculate the average cycle time of the process step. The same process is repeated for the interactions performed by the querying credit checker. He can then compare these two benchmarks to evaluate his personal performance.

“Compare with Others” form (Figure 7) shows the results of the queries. The credit checker sees that he has been stricter than the average: he has rejected 11 per cent of the requests during the last three months, whereas the average is 7 per cent. The average cycle time (19 minutes) of his credit checks is also higher than the 16 minutes of the average worker. His personal record, however, does not differ from the average in the use of specialists or in the percentage of requests that have led to credit losses or other problems. Stricter policy and lengthier inspection of requests have not increased the quality of his credit checking process. He decides to aim to reduce the processing time and accept more requests in the future.

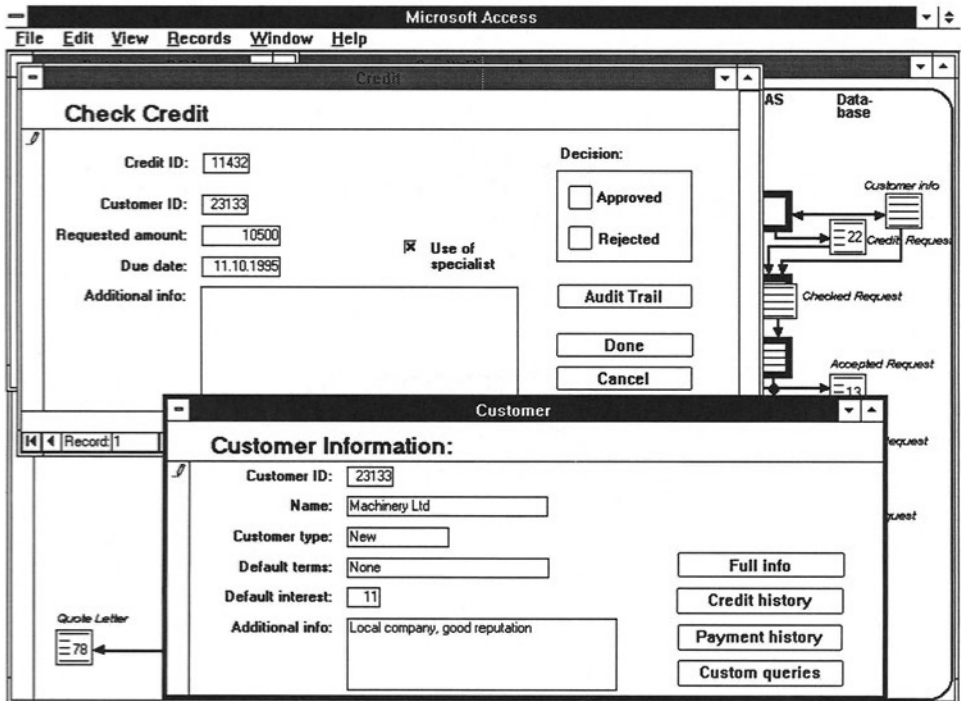


Figure 8 Credit check using the Embedded Credit Application System.

To start checking new credit requests, the agent clicks with a mouse the “Check Credit” process element of the organizational interface (Figure 6). The respective module of ECAS is executed. It opens the Credit form of ECAS and retrieves from the Credit table the next request in the queue (Figure 8). It also opens the Customer form to provide detailed customer information. A new instance is added to the Interaction table that relates the worker, the process step, and the credit request, and also tells when the interaction started and ended. Finally, the credit checker is registered as “engaged” in the Persons table.

In the following scenario, we assume that the agent faces a tricky request and needs to consult a "specialist." He clicks the concept "specialist" in the organizational interface to open the Cast form (Figure 9). Information about specialists is automatically filtered from the staff knowledge system. He can then check out which workers are specialists, which are present, and, if possible, not busy, and where the suitable workers are located. He can contact the most suitable person, possibly using e-mail (the Contact button).

The credit checker records the outcome of the check using the Credit form (Figure 8). He also fills the check box to indicate that a specialist took part in the interaction. ECAS updates the Credit table and, in the case of approval, forwards the request to wait for modification. Finally, the performance monitoring service of ReDIS stores the use of specialist, the finishing time, and the outcome of the process in the Interaction table.

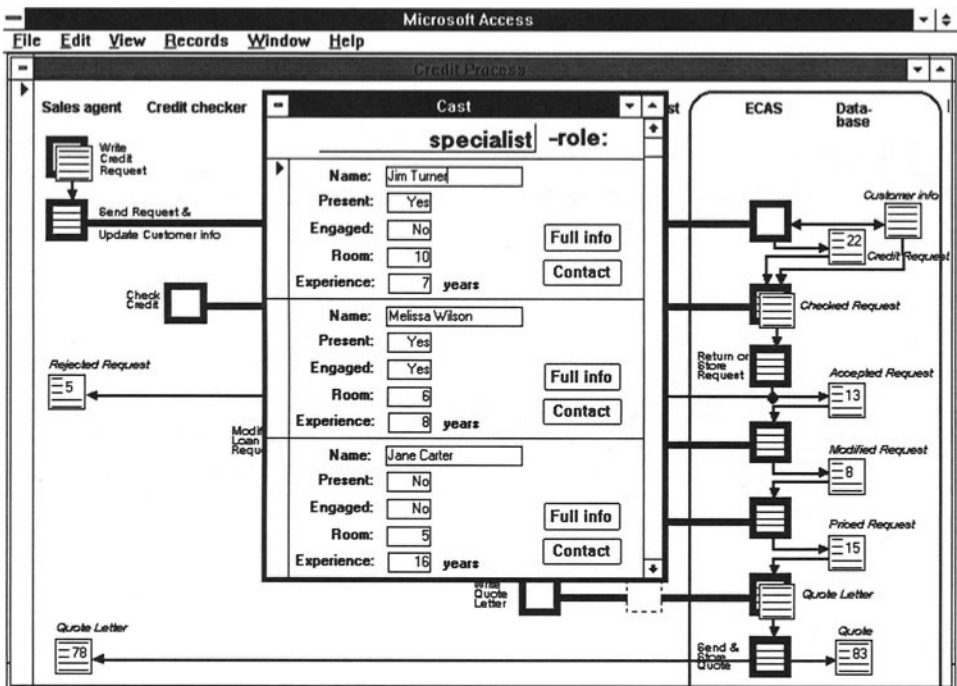


Figure 9 The cast form of the organizational interface.

The breakdown management services form the breakdown layer of ReDIS. They help agents during breakdowns quickly shift focus from routines to various details, including ECAS and the databases in which part of the work-related knowledge of agents has been encoded. The audit trail and other breakdown management services offer support during breakdowns that may emerge during the credit issuance process. After all, all situations cannot be solved simply by consulting a specialist. The audit trail service helps agents track down the life cycles of credit requests. The hyperknowledge

characteristics of ReDIS play an important role when more knowledge is needed about previously unimportant aspects of work.

We illustrate the use of breakdown management services in the following scenario: A clerk is writing a quote letter for a significant sum of money. He knows that the customer has a poor credit history and begins to wonder whether the credit request should have been accepted in the first place. Maybe an error has taken place earlier in the process. He decides to investigate before completing the letter. But how to tackle the problem? First, he needs to understand how credit requests are checked in the credit issuance process. Second, and more importantly, he needs to know who is responsible for checking this particular credit request.

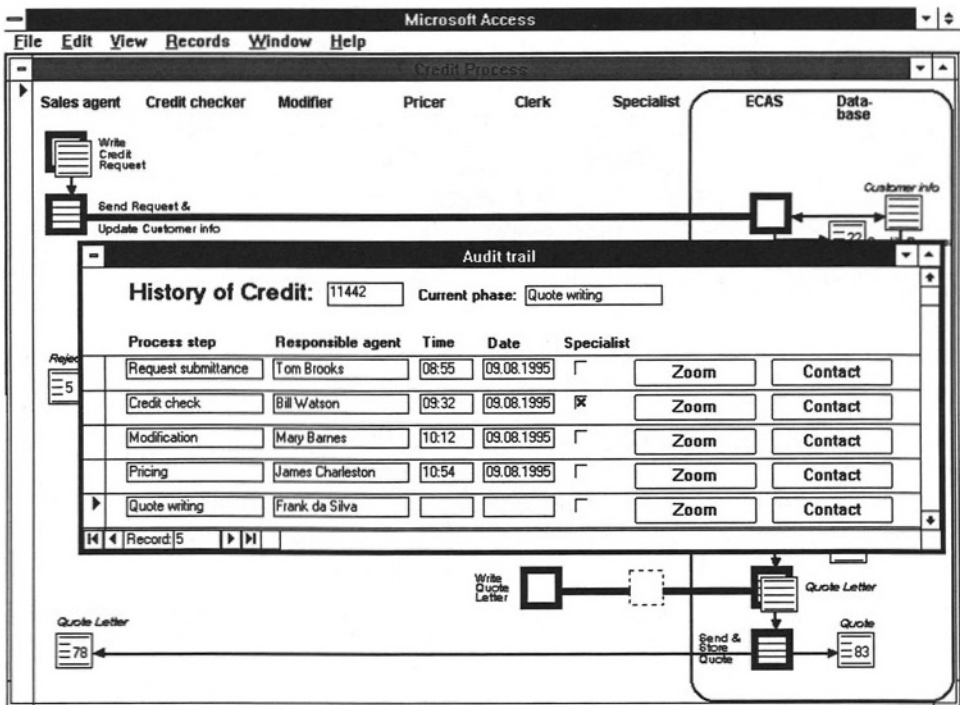


Figure 10 The audit trail form.

The clerk has internalized from the organizational interface (Figure 6) that credit checkers are responsible for checking incoming credit requests. He activates the audit trail service (Figure 8) to trace the life cycle of the credit, and zooms in on the details of the credit-check process (Figure 10). He stepwise executes the KCN model of the credit-check process (Figure 7) using the simulated enactment services of ReDIS. The original credit request and customer information are available during the simulation. The customer information confirms his doubts about the creditworthiness of the company. He sees from the KCN model how ECAS automatically forwards requests according to the decisions of credit checkers. He realizes that the request would have automatically gone to a modifier rather than being returned to the sales agent, if the credit checker had *accidentally checked*

the request approved. Convinced that the case is worthy of more effort, he contacts the responsible agent to discuss it. The KCN models of ReDIS help agents in different roles create a shared explicit language of their work routines, making the sharing of tacit knowledge easier. Fixing breakdowns is facilitated.

In summary, the services of the business and breakdown layers of ReDIS help people (1) develop a theoretical understanding of their office work, including its computerized aspects; (2) work effectively; (3) maximally exploit knowledge gained from the routine use of ReDIS to monitor and continuously improve their performance; (4) develop and test multiple hypotheses of possible causes of breakdowns to get office routines back on track fast; and (5) participate in and contribute to work redesign projects on the project layer of the Credit Corporation.

5 CONCLUSIONS AND FUTURE RESEARCH

Insufficient knowledge of work practices and the role of CBIS in enabling and conditioning these practices, especially in the lower echelons of organizations, reduces the interpretive flexibility of CBIS. CBIS with high interpretive flexibility help people master their computer-supported work as a whole, share knowledge of their work practices, and maintain constructive, internal competition that motivates the creation of knowledge. We offered Dual Information Systems, a reconceptualization of CBIS that reduces the time-space disjuncture between the development and use of information systems *vertically* by interpreting all the computerized tasks and chunks of knowledge as the work and knowledge of agents, and *horizontally* by offering agents redundant information about their inter-connected work processes.

In Section 2, we provided a theoretical backing of DIS. In Section 3, we presented the conceptual basis of DIS, augmented it with the hyperknowledge framework to develop the conceptual design of the business and breakdown layers of DIS, and proposed the use of the KCN language as a generic work, knowledge conversion, and software process modeling language to construct organizational interfaces of DIS. In Section 4, we described a part of the ReDIS prototype to crystallize the conceptual design and encourage commercial implementations.

This paper has identified several design principles for the next generation of information systems in office environments. First, to reach and maintain high interpretive flexibility of information systems, it is fruitful to focus design on three conceptual layers of DIS: the project, business and breakdown layers. Second, the design of DIS must ensure that agents in the business layer of a hyperknowledge organization can easily enact routines according to partly prescribed work flow procedures, develop a theoretical understanding of their computer-supported work, and monitor their work performance using quantitative and qualitative benchmarks. Third, the design must offer breakdown management services to help agents deepen their theoretical knowledge during breakdowns, find out what is wrong, and recover. Fourth, the design must help agents quickly return to their routines or, if recovery is not possible due to the severity of breakdowns, enter the project layer to redesign work.

Process modeling languages and hyperknowledge environments are the cornerstones of the business and breakdown layers of DIS. Work routines, including their computerized parts, can be designed using process models, encoded in hyperknowledge environments, and visualized for enactment and reflection through organizational interfaces. Agents can flexibly navigate in the hyperknowledge environment, deepen their knowledge of the specific details of work, and understand the interconnections between their work and the business process as a whole.

We have drawn on Nonaka's hypertext organization model because a dual organizational design and a dual information system design mutually reinforce each other, increasing the interpretive

flexibility of information systems. In traditional bureaucracies, (1) the creation of knowledge is fairly limited to the upper echelons of organizations (Nonaka, 1994); (2) rigid, narrowly defined jobs make it difficult for nonmanagement workers to exploit the knowledge-creating potential of hyperknowledge-based DIS (Zuboff, 1988); and (3) the performance monitoring services of DIS are likely to subject workers to unilateral managerial control. Managerial performance monitoring in hyperknowledge organizations should focus on results rather than on precise measurement of work steps to ensure the autonomy of functional work groups in the business layer (Hammer, 1990, pp. 108-112). Moreover, the extent of managerial control should be negotiated *constructively* by managers and functional work groups in the project layer, and subjected to renegotiation if managers or work groups cannot carry out their responsibilities or meet performance objectives. Käkölä (1995b) presents a more detailed analysis of the implications of interactions between managerial control patterns and coordination technologies for organizational working and learning.

Future research must establish more elaborate guidelines for the design of hyperknowledge organizations in which *all* agents can exploit the interpretive flexibility of DIS to work efficiently in the business layer, to create knowledge effectively in self-organizing project teams, and to accumulate knowledge in the knowledge-base layer. The conceptual design of bDIS must also be extended so that it covers the project layer of hyperknowledge organizations.

The next step is to assess and validate the utility of the services of business and breakdown layers of DIS by testing ReDIS first in the laboratory and later in the field. Although this research is still in progress, earlier research of the first author indicates that a system like ReDIS can be useful for working and learning. First, process model-based organizational interfaces help agents work and understand their work as a whole (Käkölä, 1996). Second, useful hyperknowledge-based working and learning environments can be built (Vanharanta, Käkölä and Back, 1995; Vanharanta, Käkölä and Kangas, 1995). Consequently, we are confident that Dual Information Systems in general, and their business and breakdown layers in particular, offer substantial benefits to organizations that implement them to increase the interpretive flexibility of information systems in office work.

6 ACKNOWLEDGEMENT

We are grateful to Inger Eriksson and Markku Nurminen, whose innovative visions about Embedded Application Systems inspired our research, and Tomas Isakowitz and the four anonymous reviewers for their constructive comments on earlier versions of this paper.

7 REFERENCES

- Bush, V. (1945). As We May Think. *The Atlantic Monthly*, 176(1), pp. 101-108.
- Bødker, S. and Grønbaek, K. (1991). Design in Action: From Prototyping by Demonstration to Cooperative Prototyping, in *Design at Work* (eds. J. Greenbaum and M. Kyng), Lawrence Erlbaum Associates, pp. 197-218.
- Boland, R.J. Jr. (1991). Information Systems Use as a Hermeneutic Process, in *Information Systems Research: Contemporary Approaches and Emergent Traditions* (eds. H.-E. Nissen, H.K. Klein and R. Hirschheim), Elsevier Science, pp. 439-458.
- Chang, A-M., Holsapple, C. W., and Whinston, A. B. (1989). A Decision Support System Theory. Working paper. University of Arizona, Tucson, USA.

- Chang, A-M., Holsapple, C. W., and Whinston, A. B. (1993). Model Management Issues and Directions. *Decision Support Systems*, 9(1), pp. 19 - 37.
- Chang, A-M., Holsapple, C. W., and Whinston, A. B. (1994). The Hyperknowledge Framework for Decision Support Systems. *Information Processing and Management*, 30(4), pp. 473-498.
- Conklin, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer*, 20(9), pp. 17 - 41.
- Ciborra, C.U. and Lanzara G.F. (1994). Formative Contexts and Information Technology. *Accounting, Management & Information Technologies*, 4(2), pp. 61-86.
- Curtis, B., Kellner, M., and Over, J. (1992). Process Modeling. *Communications of the ACM*, 35(9), pp. 75-90.
- Daft, R.L., Lengel, R.H., and Trevino, L.K. (1987). Message Equivocality, Media Selection, and Manager Performance: Implications for Information Systems. *MIS Quarterly*, 11(3).
- El Sawy, O.A. and Khorshid, H.S. (1994). A Design Theory of Virtual Workflows. Research Paper IOM 94-13, School of Business Administration, University of Southern California.
- Engelbart, D. (1963). A Conceptual Framework for The Augmentation of Man's Intellect, in *Vistas in Information Handling* (eds. P. W. Howerton and D. C. Weeks), Cleaver-Hume Press, London, 1, pp. 1-29.
- Eriksson, I., Hellman, R., and Nurminen, M. I. (1988). A Method for Supporting Users' Comprehensive Learning. *Education & Computing*, 4(4), pp. 251-264.
- Eriksson, I. and Nurminen, M. I. (1991). Doing by Learning: Embedded Application Systems. *Journal of Organizational Computing*, 1(4), pp. 323-339.
- Gasser, L. (1986). The Integration of Computing and Routine Work. *ACM Transactions of Office Information Systems*, 4(3), pp. 205-225.
- Gerson, E.H. and Star, S.L. (1986). Analyzing Due Process in the Workplace. *ACM Transactions of Office Information Systems*, 4(3), pp. 257-270.
- Hammer, M. (1990). Reengineer Work: Don't Automate, Obliterate. *Harvard Business Review*, 68, pp. 104-112.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperCollins Publishers, New York.
- Heidegger, M. (1977). *The Question Concerning Technology*. Harper & Row, New York.
- Hellman, R. (1989). User Support: Revealing Structure Instead of Surface. *Behaviour & Information Technology*, 8(6), pp. 417-435.
- Jennings, R. (1993). *Using Access™ for Windows™*. Que Corporation, Carmel, IN.
- Kogut, B. and Zander, U. (1992). Knowledge of the Firm, Combinative Capabilities, and the Replication of Technology. *Organization Science*, 3(3), August, pp. 383-397.
- Koota, K. (1995). Tietojärjestelmien käyttäminen CSCW-prosesseissa organisaation oppimisen ja tiedon luonnin tukena. Pro Gradu-tutkielma. Turun Yliopisto, Tietojenkäsittelyoppi.
- Koota, K.I. and Käkölä, T. (1996). The Knowledge Creation Net-process modeling language. Manuscript.
- Käkölä, T. (1995a). Increasing the Interpretive Flexibility of Information Systems through Embedded Application Systems. *Accounting, Management, & Information Technologies*, 5(1), pp. 79-102.
- Käkölä, T. (1995b). Designing and Deploying Coordination Technologies for Fostering Organizational Working and Learning: From Vision to Reality? Forthcoming in *Scandinavian Journal of Information Systems*, 7(2).
- Käkölä, T. (1996). Evaluation of an Embedded Application System for Supporting Organizational Working and Learning. Manuscript submitted for publication.

- Käkölä, T. and Koota, K.I. (1996). Redesigning Work with Dual Information Systems: the Work Process Benchmarking Service. Proceedings of the 29th Annual Hawaii International Conference on System Sciences, Vol. 3, pp. 461-471. IEEE.
- Lyytinen, K.J. and Ngwenyama, O. K. (1991). What Does Computer Support for Cooperative Work Mean? A Structural Analysis of Computer Supported Cooperative Work. Accounting, Management & Information Technologies, 2(1), pp. 19-37.
- Malone, T.W. (1985). Designing Organizational Interfaces. Proceedings of CHI '85 Human Factors in Computing Systems, pp. 66-71. ACM, New York.
- Nelson, T. (1987). Literary Machines. The Distributors, Indiana.
- Nielsen, J. (1990). Hypertext and Hypermedia. Academic Press Inc, San Diego.
- Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. Organization Science, 5(1), pp. 14-37.
- Nurminen, M. I. (1988). People or Computers: Three Ways of Looking at Information Systems. Studentlitteratur, Lund, Sweden & Chartwell-Bratt.
- Orlikowski, W. (1992). The Duality of Technology: Rethinking the Concept of Technology in Organizations. Organization Science, 3(3), pp. 398-427.
- Orlikowski, W. and Robey, D. (1991). Information Technology and the Structuring of Organizations. Information Systems Research, 2(2), pp. 143-169.
- Peterson, J.L. (1981). Petri Net Theory and the Modeling of Systems. Prentice Hall, Englewood Cliffs, NJ.
- Rein, G. (1992). Organization Design Viewed as a Group Process Using Coordination Technology. Ph.D. dissertation. MCC Technical Report CT-039-92, Austin, Texas.
- Roos, L. L. Jr. and Starke, F. A. (1981). Organizational Roles, in Handbook of Organizational Design, 1: Adapting organizations to their environments (eds. P. C. Nyström and W.H. Starbuck), pp. 290-308. Oxford University Press, Oxford.
- Spendolini, M.J. (1992). The Benchmarking Book. American Management Association, New York.
- Stryker, S. and Statham, A. (1985). Symbolic Interaction and Role Theory, in Handbook of Social Psychology, 1 (eds. G. Lindzey and E. Aronson), Random House, New York.
- Swanson, E. B. and Ramiller, N.C. (1993). Information Systems Research Thematics: Submissions to a New Journal, 1987-1992. Information Systems Research, 4(4), pp. 299-330.
- Tyre, M.J. and Orlikowski, W.J. (1994). Windows of Opportunity: Temporal Patterns of Technological Adaptation in Organizations. Organization Science, 5(1), pp. 98-118.
- Vanharanta, H., Käkölä, T., and Back, B. (1995). Validity and Utility of a Hyperknowledge-Based Financial Benchmarking System. Proceedings of the 28th Annual Hawaii International Conference on System Sciences, Vol. 3, pp. 221-230. IEEE.
- Vanharanta, H., Käkölä, T., and Kangas, K. (1995). Usability of a Hyperknowledge-Based Executive Support System for Financial Benchmarking. Proceedings of the 28th Annual Hawaii International Conference on System Sciences, Vol. 3, pp. 130-139. IEEE.
- Winograd, T. and Flores, F. (1986). Understanding Computers and Cognition: A New Foundation for Design. Ablex Publishing Corporation, Norwood, NJ.
- Wynn, E. (1991). Taking Practice Seriously, in *Design at Work* (eds. J. Greenbaum and M. Kyng), Lawrence Erlbaum Associates, pp. 45-64.
- Zuboff, S. (1988). In the Age of the Smart Machine - The Future of Work and Power. Oxford: Heinemann Professional Publishing Ltd.

8 BIOGRAPHY

Timo Käkölä is a research scientist at the Department of Computer Science and Information Systems, University of Turku, Finland. His research interests include IT-enabled organizational designs for effective organizational creation and sharing of knowledge, and the integration of collaborative technologies such as workflow, hyperknowledge, and group support systems into Dual Information Systems that support work and work redesign. He has published a number of articles in leading scientific journals and conferences.

Kalle Koota is a systems analyst in the R&D department of Orion Corporation, Finland. His research interests include computer supported cooperative work, process simulation, and process modeling. His practical interests include document management and IT support for shared knowledge creation.