Fostering Organizational Learning with Embedded Application Systems: The XTEND2 Prototype

Timo Käkölä
Department of Computer Science and Information Systems
University of Turku, DataCity, FIN-20520 Turku, Finland

Abstract
In this paper it is argued that agents must understand their work as a whole, including computerized tasks, if they are to employ information systems for fostering organizational learning. The Embedded Systems Approach is proposed for redesigning the structure and the components of software systems so that it is easier for agents to develop such an understanding. In the approach the applications are embedded in extended support systems that make the organization of work and the coordinating role of information systems explicit as well as support agents in handling unexpected breakdowns. The resulting systems are called Embedded Application Systems. The XTEND2 prototype is described to illustrate the approach and to inspire commercial implementations.

1. Introduction

Much attention in information systems research has been paid to developing systems on the one hand for enhancing managerial decision making and learning (e.g., group and executive support systems) and on the other hand for automating and coordinating routine work on the shop floor (e.g., workflow systems). For example, in a recent study on information systems research thematics, Swanson and Ramiller [38] found that researchers have focused extensively on computer supported cooperative work as well as managerial decision making, and established strong links between these two areas. While this research continues to be relevant, it undermines the need for new knowledge with regard to the development of systems that tap the resources and capabilities of all organizational members. Even Senge [36], while using the phrase "organizational learning", focuses on managerial learning and the use of computer-based information systems (CBIS) in fostering managerial learning.

It is argued in this paper that information systems research must focus on enabling organizational learning, not only the learning and decision making of the middle managers and executives. Because the agents on the lower echelons primarily employ transaction processing and other production systems, researchers need to produce new knowledge for redesigning these systems so that they would serve as flexible working and learning environments for facilitating individual and organizational learning instead of merely automating routine work.

Redesigning presupposes changes in the technical and conceptual structure of information systems. Rathwohl [34, p. 436] asks: "In what ways does the current structure of information technologies facilitate our ability to learn - to acquire more useful images of the world, to ask better questions, and to develop a more enlightened basis for practical action?" Our experiences in the Knowledge and Work (K & W) project as well as the experiences of Zuboff [40], Orlikowski [31], [32] and many others indicate that too often the structures of CBIS provide poor support for learning. This is largely because they do not allow the human agents using them to see the constructed nature of CBIS [32]. For example, most transaction processing and workflow systems do not explicate (1) the work context and (2) their coordinating role in this context thereby making the computer-mediated communication and coordination invisible [7], [31].

In our earlier research we have employed structuration theory [11] to understand how organizations learn [17] and the Human-scaled Information System (HIS) perspective [30] to search for answers to the redesign issue [8]. Drawing upon this theoretical background and our experiences in the XTEND project2 we propose the Embedded Systems Approach (ESA) as a way of redesigning information systems so that they foster organizational coordination and learning as well as the handling of unexpected breakdown situations. By applying this approach the organizations are able to develop Embedded Application Systems [9] that change the conceptual structure as well as the components of software systems so that they provide agents with such knowledge that the agents are able to see the constructed nature of CBIS and thereby monitor all aspects of their work, including the way information systems and institutional structures are combined within their work context, and how this combination by providing resources and rules simultaneously enables and conditions their actions [24],

---

1 The Knowledge and Work project (1985-1989) at the University of Turku was led by Professor Markku I. Nurminen and was mainly financed by the Academy of Finland.
2 The XTEND project (1988-1990) at the University of Turku was led by Professor Markku I. Nurminen, and was mainly financed by the Finnish Work Environment Fund.
This knowledge is a prerequisite to employing CBIS for fostering organizational learning; without it the agents are not able to be in control of and responsible for their work as a whole, including the computer-mediated parts of work [30], [40].

The rest of the paper is structured as follows. In Section 2 the Embedded Systems Approach (ESA) is presented. In Section 3 the XTEND2 prototype is described to illustrate the approach and to inspire commercial implementations. In Section 4 the conclusions are stated and issues for future research are discussed.

2. The Embedded Systems Approach to Redesigning Information Systems

In Section 2.1 we present some knowledge requirements for the effective employment of the CBIS. In Section 2.2 we review the lessons learned in the XTEND project where CBIS prototypes trying to meet these knowledge requirements were developed. Drawing upon these lessons as well as the conceptual model presented by Eriksson and Nurminen [9] we propose in Section 2.3 the Embedded Systems Approach for designing Embedded Application Systems [9] that provide agents with knowledge about the organization of work, task coordination and the coordinating role of information systems. Organizational and software process models constructed with the help of Role Interaction Nets [35] are proposed as the structural basis of Embedded Application Systems in Section 2.4.

2.1. Knowledge requirements in computer-mediated work environments

The need for fast organizational learning combined with the need to employ CBIS for fostering learning pose new requirements on agents' competence [19]. What kind of knowledge is needed to fulfill these requirements? Anderson and Sharrock [1, p. 158-159] give a partial answer by stating: “Social life could not begin if, first, we had to align bodies of knowledge, frames of reference, meaning structures. Rather what is presumed is procedural knowledge - how to find things out in this setting, for these purposes etc. What social competence consists in is the knowledge of how to ask questions, pick up, locate, see at a glance, and all the ‘normally thoughtless’ looking and seeing that we all do day in and day out.”

Unfortunately agents often lack parts of this knowledge in computer-mediated work environments [3], [7], [40]. They do not understand (1) the nature of the social practices as a whole and how these practices are articulated in time and space by the structural properties of organizations, (2) their own role in the organization, and (3) the role of information systems in mediating communication, collaboration, and control in the articulation processes. One major reason for this is that most information systems have a technical and conceptual structure that (1) separates symbolic information from the material and social systems the symbols represent, (2) hides the processing rules and retention structures into the software and database schemas respectively, and (3) blurs the role of human agents as the producers and consumers of information [4], [30], [32].

This lack of a deep understanding of the computer-mediated work has several implications on information systems development and use. The agents face considerable difficulties in monitoring their actions, since they are not fully able to interpret and validate the meaning of information produced by the systems, and they cannot see and experience the outcomes of their computer-mediated actions [40, pp. 79-96]. Even if the agents were able to produce varied interpretations of computerized data, they could not question the basis on which the data has been constructed [22]. Managing exceptional or new situations is difficult in this situation [7]. The agents may not be able to take the responsibility for their work as a whole, including computerized tasks, because their abilities to control all aspects of work are limited. Their actions are also more likely to result in unintended outcomes such as coordination breakdowns, and their capabilities of intervening in the existing social practices are endangered [14].

The conceptual structure of software systems must be fundamentally redesigned if the agents are to enact collective meanings for the information produced and communication mediated by the systems [20] as well as develop such procedural knowledge that they can monitor and consequently be responsible for their computer-mediated actions. First the systems have to enable the agents to develop shared stocks of knowledge concerning the planning, coordination and articulation of work processes [7], [36], [40]. Secondly they must help the agents to understand the internal structure of software, if the agents are to operate skillfully through the information system and to use the system as a source of learning and feedback [40, p. 73]. Finally, the systems must also be tailor able so that they facilitate the improvement of work practices.

2.2. The road towards the Embedded Systems Approach

In the XTEND project the author played a key role in designing and implementing a prototype with an extended support system that aimed at reinforcing organizational learning by providing agents with shared stocks of knowledge concerning their work context as well as the internal structure of the applications [8]. Unfortunately the system did not fully meet our expectations, because we did not adequately question the feasibility of the commonly accepted components of software systems and their relations. The components are (1) applications performing the production function; (2) support systems assisting agents in understanding the function of the applications; and (3) human-computer interfaces [23].

The extended support system had a three-layered structure: (1) an organizational model that visualized the organization and coordination of work and allowed the users to navigate in the task lattice of the organization [10]; (2) a menu structure allowing the users to perform...
the specific computerized sub-tasks of each work task in the task lattice in a simulation mode, view the databases, etc.; and (3) the application implementing the computerized sub-tasks in the simulation mode. This structure was similar to the structure of traditional support systems except for one crucial difference. If the agent did not want to experiment with other agents' tasks in the other work units, she could experiment with the application that was a copy of the one from which she had started the support system. However, this time the application was embedded in the support system and enhanced with some added features specific to the support system.

While the structure seemed reasonable at first and sufficiently in line with the traditional approaches of developing support systems but yet crucially different, a number of difficulties soon emerged. We ended up having two slightly different versions of the application system: one from which the support system was invoked and the other embedded in the support system itself. As a result, maintaining the software proved troublesome. Moreover, the prototype became complex to use, since the users had to become proficient in navigating between the two systems as well as within the support system as a whole.

Clearly the structure included one application system too many. The solution in line with the traditional wisdom would have been to eliminate the one embedded in the support system. However, Eriksson and Nurminen [9] proposed a radically different conceptual solution which, when applied in practice, would result in the removal of the application from which the support system was invoked, and thus only the application embedded in the support system would remain. They called such applications Embedded Application Systems. Next we take a closer view on what such a system could look like conceptually by drawing upon our experiences as well as the solution presented by Eriksson and Nurminen [9].

2.3. The Embedded Systems Approach for redesign

According to the Embedded Application System concept the applications are embedded into and enacted from an organizational model. This model serves as the building block of an extended support system that has four central characteristics. (1) It eliminates the boundaries between work and information systems by viewing applications as structural resources and rules that the agents draw upon in their knowledgeable interactions. (2) It facilitates learning before doing by making explicit who is responsible for and who is dependent on which work processes (including their computer-mediated parts), and what are the cause-effect relations between various process fragments. This is realized by modeling and visualizing the various units of the organization, the different roles of agents, the tasks and sub-tasks (including the computerized ones) belonging to these roles, the documents produced and used in these tasks, and the databases [8]. Such knowledge is the basis for reconstructing the missing link between CBIS and work systems [7], [9], [30]. (3) It allows the agents to execute computerized tasks (i.e. algorithms) step by step and check the results from the databases employed by the algorithms. Each execution step fulfills the basic requirement of an algorithm: it is always so "elementary that it can be performed by a human using just pen and paper, or by a machine" [6, p. 7]. This is especially helpful in breakdown situations that cannot be prescribed in advance. In these situations the agents need to zoom in the details of their work to determine what is wrong and to resolve the situation. (4) In a special organizational learning mode it allows the agents to instantiate roles that do not belong to their own areas of responsibility, use databases identical to those in use but not live, and search these databases in an ad-hoc fashion. This approach is an extended computerized version of the group learning technique called "the redundancy of functions" typically found in autonomous work groups [28, p. 99]. The technique increases the flexibility of the work organization by involving cross training and upgrading knowledge so that each agent is able to take many or even all the roles in his or her group. However, ESA allows flexible role taking even among the agents operating in different organizational units.

This conceptual structure fundamentally transcends the traditional concepts of human-computer interface and human-computer interaction by focusing on how agents act, interact and learn by drawing upon various structural resources and rules, and thus partly sustaining and partly modifying the prevailing social structure. We therefore will subsequently employ the term "organizational interface" [25] in referring to this interface through which the agents act and interact. Furthermore, we call the design approach for developing Embedded Application Systems the Embedded Systems Approach (ESA).

2.4. Process Modeling and the Embedded Systems Approach

How can organizational interfaces be developed so that they would meet the requirements outlined above? Since Eriksson and Nurminen [9] focus on broad conceptual work, their presentation does not address this question. Moreover, their ideas are built closely on the hierarchical menu-based interfaces employed in the XTEND prototype. However, it seems that totally new ideas would be needed to build learning enhancing Embedded Application Systems. Indeed, when pursuing ESA a dilemma is faced: most of the traditional structured and object-oriented systems analysis and design approaches are poorly suited to developing organizational interfaces, because (1) they focus almost exclusively on modeling data flows, data structures and other components of technical systems and (2) they are difficult to understand and employ by agents without specific training in computer science [37], [5]. We argue on the basis of the statements (1) and (2) that these modeling languages reinforce the development of reified and objectified information systems over which most agents retain little control.

However, a new class of languages called process
modeling languages has recently emerged. These languages remedy a number of the weaknesses of the technically oriented modeling languages. Curtis, Kellner and Over state [5, 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.” What is more important from our point of view, process modeling unites organization and systems design by focusing on (1) “interacting behaviours among agents, regardless of whether a computer is involved in the transactions” [5, p. 75], (2) the agent’s behaviour at the interface, and (3) the flow and transformation of data within the software system. Furthermore, process models provide agents with a systemic understanding of work, by answering all or most of the following questions [5, p. 77]: “what is going to be done, who is going to do it, when and where will it be done, how and why will it be done, and who is dependent on its being done”. Still, little research has been done on the use of process modeling languages in designing organizational interfaces.

Kikkilä [18] argued for the employment of the Role Interaction Net (RIN) process modeling language [35] in ESA. The language was originally developed for the purpose of organization design [35]. A RIN is composed of a set of concurrent roles. The behaviour of a role is described by its solitary actions as well as its interactions with other roles. Roles and interactions serve as simple but expressive primitives for modeling both the structural and process aspects of organizations. Techniques from Petri nets [33] are utilized to give the language the process description and enactment capability. The enactment capability means that in a proper enactment environment the RIN models can be directly executed [35]. This is of vital importance to ESA: contrary to static descriptions, executable models provide organizations with a clear incentive to maintain them. The RIN models are consequently appropriate to serve as the structural basis of Embedded Application Systems. Figure 1 clarifies a few basic notations of the RIN Language. In the following Section the use of Role Interaction Nets in ESA is illustrated with the help of the XTEND2 prototype.

3. Description of the XTEND2 Prototype

The XTEND2 prototype was developed for testing the relevance of our ideas. It simulates the inventory management system (IMS) of an organization in food industry. The organization had been analyzed during the Knowledge & Work project [7]. It had two inventory units: the bulk and the buffer serving as the long-term and the short-term stocks respectively. The material was supplied by three manufacturing and packing (M/P) units and delivered to customers when requested. Therefore the M/P units and the customers were also relevant to designing the prototype. The workers in the inventories handled and transported food pallets within and between organizational units with fork lifts. The IMS was used to keep track of the pallets so that they could be stored in suitable places and retrieved quickly when needed.

![Figure 1. Some notations of the RIN Language](35, p. 162).

The IMS and the organizational context give ample opportunities for research with regard to the development and use of Embedded Application Systems for supporting organizational coordination and learning. On the surface the operations of the organization seemed deceptively simple, but a more detailed analysis revealed that exceptional or erroneous breakdown situations took place frequently within and between the units [15]. The IMS caused a number of these situations indirectly. For example, the agents often entered erroneous information to the system, because they did not understand how the system mediated task coordination between the units. Moreover, they received little support from the system for handling or recovering from these situations.

The IMS was no longer in use. Consequently the solutions of any existing system were not a constraint. The project was therefore decided to carry out in a laboratory environment. A hypermedia-based development platform was a necessity, since the organizational interfaces have to facilitate flexible navigation in the systems without making them too complex and unusable. Macintosh Quadras and the HyperCard system [12] were selected as the platform. In the following the redesign of the IMS in the XTEND2 prototype is presented.

3.1. The redesign of an Inventory System as five Embedded Application Systems

Two aspects of the inventory organization are especially helpful in redesign. First, the five units in the inventory organization are relatively autonomous. Second, there is little division of labour between the agents working in the same unit. In this case the integrated IMS can be divided logically into five Embedded Application Systems: one for each of the three M/P units and one for each of the two inventories [15]. Hereby the agents are given full responsibility not only for the material objects but also for the information. There are two principal factors in the redesign that make it easy for the agents to monitor their actions. First, each system is smaller than the integrated IMS and thereby easier to manage. Second, the clear organization of computer-mediated work facilitates...
coordination between the units [15].

The RIN models serve as the organizational interfaces of Embedded Application Systems in the XTEND2 prototype. The agents enact their computerized tasks from the interface assigned to them. They can also use the interface as an extended support system as described in Section 2.3. In Sections 3.2 and 3.3 a working scenario is employed to illustrate how the prototype fosters organizational learning and the handling of breakdown situations. The agents using the system are assumed to be relatively ignorant about their work context and have limited computer experience. This assumption reflects the competence of the agents in the original organization.

3.2. Fostering organizational learning in XTEND2

The organizational interfaces of all five Embedded Application Systems have been implemented by employing the RIN view defined as the union of the set of roles and interactions that all agents in an organizational unit share and the set of roles in other units that interact with the roles in this unit directly. The view is called a domain RIN.

Figure 2 shows the domain RIN creating the organizational interface for the Embedded Application System of the bulk. Each agent in the bulk can instantiate the "bulkman" role and then enact several computer-supported work processes by clicking with a mouse the respective interaction parts below the role name. The parts visualize the processes the agents in the role are responsible for. However, the application itself is not visible in the domain RIN. It has been embedded in the work processes of the bulkmen, because in ESA the information and work systems are viewed to be inseparable. Only the database is visible, because it provides a part of the organizational memory necessary for coordinating the work processes [39].

In the first process a bulkman receives food pallets and information on each pallet from a packer in an M/P unit, uses the application to determine the shelf positions for the pallets, and finally stores the pallets to the positions. In the second and third processes bulkmen are responsible for checking that all the pallets are positioned correctly in the inventory as well as preserving the quality of the goods. Since these work processes are not related in a specific temporal order, they are described using a choice construct visualized by the rounded box. In the fourth and fifth processes the bulkmen receive delivery orders from the buffer-men, use the system to find out which pallets are needed and where they are located, fetch the pallets and finally deliver them to the buffer. The sixth process involves returning some pallets to one of the M/P units.

The domain RIN models the structuring of social (inter)actions in time and space. The notion of space is inscribed into the roles. The temporal structuring of activities is visualized by the order of the representation. Furthermore, the role of the Embedded Application System in enabling these temporally and spatially situated work practices through shared databases is clarified. For example, the Database role in Figure 2 depicts how the shelf position and pallet information mediate coordination between the processes the bulkmen enact in their daily work. The intricate combination of both the system and the institutionalized structures is consequently illustrated to the agents.

The organizational interface allows the agents to monitor the results of each process enactment. They can start browsing the database simply by clicking the corresponding database interaction part with a mouse. In that case they are shown a detailed view of the instances of all the entities created or deleted in the enactment. They can also see the updated attribute values for each updated instance. Each enactment can be rolled back if unwanted

Figure 2. The domain RIN of the bulk inventory.
results are found.

The domain RIN alone may be insufficient to eliminate coordination breakdowns. But the RIN makes it easier to generate hypotheses about their possible causes. For example, when a bulkman is receiving a set of pallets from an M/P unit knowing that there is room for them in the bulk, and the computer informs that there is not, something has gone wrong. Three valid hypotheses can be deduced immediately from the RIN in Figure 2: some pallets have been (1) disposed earlier, (2) delivered to the buffer, or (3) delivered to one of the M/P units, but the corresponding changes in the database have not been done.

We now assume a scenario where a bulkman has received an order from the buffer and delivered the needed number of pallets there. The bulkman wants to know what happens next in the buffer. There are a number of good reasons for this. For example, the analysis carried out in the Knowledge & Work project revealed that the pallets were often sent from one unit to another without having proper information attached, because those sending the pallets were not aware of the importance of the information [15]. The agents receiving the pallets could not use the IMS to handle them without adequate information. To alleviate such problems the XTEND2 prototype implements the organizational learning mode (Section 2.3) allowing the agents in each unit to access the organizational interfaces in the connected units. The bulkman now starts this mode and takes the role of a buffer-man by clicking the buffer-man's "Deliver pallets to Buffer"-interaction part in the interface (Figure 2).

![Figure 3. The Inter-Unit Connections RIN for fostering organizational learning.](image)

Figure 3 shows the Inter-Unit Connections RIN that is always visible in the beginning of the organizational learning mode. This RIN visualizes the most important processes between the units to give agents an overall view of their articulation. The bulkman can deduce from the RIN that a production worker has delivered products and product information to a packer who has packaged the products into pallets. The packer has then decided based on the product information whether the pallets need to be placed in a quarantine to wait for laboratory inspection or sent to the bulk or directly to the buffer. All quarantine pallets have been delivered to the bulk or the buffer when a laboratory worker has approved the delivery. The highlighted interaction part now informs the bulkman that the next logical process for the buffer-man is to receive the delivered pallets. By clicking the RIN the bulkman enters the buffer's domain RIN that is similar to the one in the bulk. She can now receive the pallets in the buffer, check the resulting changes in the databases of the buffer's Embedded Application System, and even deliver the pallets to the customer. However, all the processes are rolled back when she returns to her own domain RIN. This design allows organizational learning without disrupting work in any unit.

However, the design principles of ESA as well as the notion of organizational learning in general do not imply that all information should be public [16]. If a requirement like this were enforced, it would lead to information overload in all units as well as to the lack of privacy and autonomy. Anyhow, each unit should have access to all information that helps in recovering from coordination breakdowns as well as to information that is directly relevant to its work planning and preparation, because this helps to eliminate breakdowns in advance [15].

3.3. Handling breakdowns in XTEND2

Paradoxically, in trying to build user-friendly systems the designers often leave users helpless in breakdowns [6, p. 116]. Effective support for breakdown handling is therefore an important determinant of high quality
software. This paradox can only be dealt with by understanding how breakdowns take place. Agents routinely monitor their actions and thus may not even be aware of them. But in breakdown situations these actions become visible and agents need to raise their level of consciousness in order to explain and to get over them. It is therefore crucial that software designs facilitate this shift in the level of consciousness and enable rapid recovery by making visible the actions normally hidden in the software.

Work practices are complex and can never be fully prescribed by normative models, because the agents are always able to act unexpectedly and their actions may yield unintended consequences. Instead of trying to build all-encompassing RIN models, the Embedded Systems Approach provides a more generic design approach to dealing with breakdowns: (1) freezing the involved databases to a state just before the breakdown, and allowing the agents to (2) reconstruct the situation by enacting the algorithms manually and (3) correct it by using the functionality of the system when the cause of the breakdown has been clarified. Hence, the prototype has no visualized (inter)actions for the breakdown situations.

XTEND2 employs RINs to help agents learn the structure of the software when breakdowns occur or the agents use the system for the first time. In the following scenario it is assumed that both situations occur, but the system would be used almost similarly in the second situation alone. Now the bulkman has to receive pallets 43 and 44 from an M/P unit by employing the organizational interface (Figure 2). Figure 4 shows the resulting Bulk-Operations RIN visualizing the corresponding part of the internal structure and procedural reasoning of the system. The RIN is automatically stepwise executed. Two interactions marked by the “\(^v\)”-check have been completed already. In the first one she has physically received pallets from a packer. This in turn has triggered a second interaction where the bulk's database has been updated accordingly. This interaction has been visualized in more detail (Figure 5), because it is a macro description.

**Table 1**

<table>
<thead>
<tr>
<th>Bulkman</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pallets</td>
<td>Inventory</td>
</tr>
<tr>
<td>Products</td>
<td></td>
</tr>
<tr>
<td>Categories</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Operations view: Receiving pallets

Figure 5 shows a role-centric RIN where both the dialogue and the functional components of software are assigned in the bulkman role to clarify who is responsible for them. The RIN also elaborates upon the structure and the use of the bulk database. In line with ESA the role-centric RINs are always detailed enough to allow manual execution of computerized tasks. This RIN reveals that the bulkman has first given the palletID codes corresponding to the received pallets. Using these codes the software has checked from the shared Pallets database that the pallets belong to the bulk, updated the status of each received pallet to reflect this belonging, counted the amount of food boxes for each product stored in the pallets, and increased the inventory quantities database accordingly. After this the control has returned to the operations RIN.

In the third interaction of the operations RIN (Figure 4) the bulkman has made a choice of allowing the computer to allocate suitable shelf positions for the pallets, and this is shown by the highlighted interaction part. Now another role-centric RIN visualizes the allocation process. The RIN resembles the one in Figure 6 but it does not show any highlighted database interaction parts or buttons on the top of the interface screen. The software first checks which product is stored into each pallet. For each product there are one or more suitable shelf categories, and for each category there are one or more suitable shelf positions. Thus the second and the third interactions find the shelf positions suitable for the products from the Products and Categories databases. Next, the software tries to find vacant positions by removing the allocated ones from the suitable ones. However, according to the Allocated Positions database there is not an adequate number of vacant positions in the bulk (Figure 6). Thus the execution of the interactions is halted.

XTEND2 now invokes the breakdown handling mode as illustrated in Figure 6. This mode deepens the agents' understanding of the algorithm where the breakdown occurred and the databases the algorithm relies on so that
the agents are able to physically check and compare the actual status of the inventory with the status represented by the databases, and then fix possible errors. The bulkman is now able to reconstruct the breakdown situation, i.e. (1) to find out what the suitable positions for the pallets are, (2) to physically check whether all of them really are occupied, and if that is not the case, to (3) determine what is wrong with the database and (4) finally correct the situation. The first four database interaction parts have been highlighted to guide the bulkman in this process. She has chosen the first interaction part and then browsed the Pallets database to find the productIDs for the pallets 43 and 44. This database accumulates a stratified record of work executed on each pallet from the time of production to the customer delivery. Visualizing this record allows the agents to internalize shared and contextual knowledge about how the work is done in the organization, what has been done and what yet needs to be done [1]. While only the ProductIDs are of interest here, learning the broader contextual knowledge may be helpful in many other breakdowns.

The bulkman is now able to physically check whether the pallet contains the product 1104. In this scenario it does and she can continue the reconstruction by searching for the suitable positions from the Products and Categories databases and by physically checking whether these positions really are occupied. She notices that they are not, returns to the terminal, and finds out from the Allocated Positions database that it incorrectly states that there are pallets in these positions. Next she learns from the Pallets database that these erroneous pallets have not been delivered anywhere from the bulk. If this is true, the most likely cause for the breakdown is that another bulkman has destroyed the pallets but forgotten to update the Allocated Positions database accordingly. After possibly discussing about this hypothesis with fellow workers, the bulkman can now return to the domain RIN, destroy the erroneous pallets, and then finish the receiving of pallets 43 and 44.

![Diagram](image-url)

**Figure 6. Handling the breakdown in the allocation of shelf positions**

### 3.4. Technical implementation considerations

From the designers' point of view the Role Interaction Nets proved relatively simple to learn and employ to visualize organizational processes. However, it is easy to create poor interfaces and designs with them. For example, the RIN Language uses arbitrary icons for presenting the interaction parts. An arbitrary icon has a meaning (e.g., the interaction part is automatic and not described in more detail) but cannot be interpreted any further [21]. Thus these icons are well suited to generic modeling and presentation purposes. Moreover, by deploying arbitrary icons the designers avoid semantic errors due to false analogies between human systems and the presentations of these systems in the interface. On the other hand, the deployment of these icons poses severe pressures on the designers' ability to create meaningful textual descriptions.
for each interaction because the icons convey few meanings. In our experience the texts often became too long and dominating but the efforts to shorten them in turn reduced their descriptive power.

Excellent maintainability of information systems is of utmost importance in learning organizations. This poses a great challenge on the Embedded Systems Approach, since the approach adds a new layer of complexity into the systems architecture. ESA calls for explicit integration of organizational and information systems design: computer applications are seen in the organizational interfaces as the means of performing computer-supported work processes belonging to human agents in the institutionalized roles. This implies that changes in the work and information systems must immediately be reflected in the interfaces. Clearly very powerful software tools are required to facilitate these change processes.

Unfortunately Hypercard is not suitable for building maintainable systems. The reasons for this can be understood by comparing the features of the tool to the requirements of the Dexter hypertext reference model [13]. The model states that a hypermedia system must be divided into three layers separated from one another. Runtime layer handles the presentation and interaction aspects of the system. The storage layer glues together the node and link components to constitute hypermedia networks. The within-component layer deals with the data contents and structure within the components. Hypercard, on the other hand, merges all these layers. For example, each domain RIN is physically stored on a card. However, the card serves as a presentation structure for the interface as well. The enactment of the processes from the interface as well as the navigation between various Embedded Application Systems is made possible by links physically stored and spatially located on top of the interaction parts of the domain RIN. Any changes in the RIN must be accompanied by the corresponding changes in the locations of the links. Furthermore, there is no linear program structure. The program code enabling both the navigational and the enactment aspects is scattered over links, cards and other components of the prototype. As a result, the code is strenuous to document and comprehend, and the cooperative work necessary for implementing this kind of a fairly large prototype is hard to carry out.

4. Conclusions and Future Research

In this paper it has been presumed that the agents working in computer-mediated work contexts often lack adequate knowledge of their work practices and the role of CBIS in enabling and conditioning these practices. The insufficient level of knowledge especially in the lower echelons of the organization in turn hampers the organization in employing the CBIS for supporting organizational coordination and learning. This paper has also argued that information systems researchers should not overlook the importance of learning on the lower echelons of the organizations and the respective need for learning-enhancing production systems on the shop floor.

In Section 2 the Embedded Systems Approach was proposed as a way of designing Embedded Application Systems [9] that enable learning before and through doing as well as support the handling of unexpected breakdowns. The Role Interaction Net Language was proposed as a generic modeling language for constructing organizational interfaces that help the agents working within and among various units to internalize shared stocks of knowledge with regard to their computer-mediated work. In Section 3 the XTEND2 prototype was described to illustrate the design approach and to inspire commercial implementations.

An enticing topic for future research is to find out whether ESA should be complemented so that role-centric data models would also become a part of the organizational interfaces. This might be beneficial, because process models alone may be insufficient to fully clarify who is responsible for which data elements and attributes of the database of each Embedded Application System. Yet, clear organizational comprehension of these areas of responsibility is vital to maintaining the quality of computer-based organizational memory as well as to handling breakdowns [9], [30].

Two other complementations to ESA are also worth investigating. First, Embedded Application Systems must inevitably be augmented by systems enabling direct communication and dialogue through which the agents are able to share and refine their interpretations. This is because organizational coordination and learning are dependent on dialogue through which the agents negotiate and enact their roles and interactions, and thereby maintain, elaborate, and modify social institutions [2]. Second, in order to foster this maintenance, elaboration, and modification the systems must be radically tailorable [26]. Radically tailorable systems allow skilled agents to create new applications by progressively modifying a working system [26]. The Oval system [27] seems an excellent approach towards this direction. The organizations will be close to an ideal computer-based working and learning environment when the work practices and the role of information systems in these practices are explicated by the RIN models, reflection and negotiation of these practices are facilitated by the dialogue-oriented systems, and the improvement of these practices as a result of the negotiations is enabled by the radically tailorable systems.

Validation of the Embedded Systems Approach is a challenging endeavour for future research. It requires not only prototyping but extensive laboratory and field testing as well. We are currently applying ESA in an office environment to test the generalizability of the approach. Although this research is still in progress we are confident that well-coordinated work combined with the improved employment of the CBIS even on the shop floor level may bring about substantial benefits to those organizations that build on our approach in redesigning systems for fostering organizational learning.
Acknowledgements
I want to thank Markku I. Nurminen for his innovative ideas and visions that initiated the research presented in this paper, and Sari Sinkkonen for her invaluable work on the XTEND2 prototype. I also thank Aimo Törn and the anonymous reviewers for their helpful comments on an earlier draft of this paper.

References