Super Users and Local Developers: The Organization of End User Development in an Accounting Company

Anders Mørch, Hege-René Hansen Åsand

Prepublication version of Åsand, H.-R., and Mørch, A.I. (2006). Super Users and Local Developers: The Organization of End-User Development in an Accounting Company. Journal of Organizational and End User Computing, 18(4), pp. 1-21.

Super Users and Local Developers: The Organization of End User Development in an Accounting Company

Hege-René Hansen Åsand and Anders I. Mørch
InterMedia, University of Oslo, Norway
Email: {hegerena, anders.morch}@intermedia.uio.no

Abstract

The paper presents a case study following the activities of super users and local developers during the adoption of a new business application by an accounting firm in Scandinavia (referred to as the Company). The Company launched a program to train super users to help with this process because of the complexity of the new system, a generic, multi-purpose application system replacing several older, non-integrated systems. The system, Visma Business (VB) is a comprehensive financial and accounting application delivered as a set of components that need to be configured for domain-specific tasks, depending on the clients the accountants will interact with. The super users and the local end user developer (also called the application coordinator) were asked to take part in this study. We documented their activities empirically and analytically, using interviews to gather data and drawing on aspects of Activity Theory for the conceptual framework for analysis. Our findings provide insight into end-user development (EUD) activities with VB: what roles were created by the Company, what roles emerged spontaneously during the process, what the various user groups (regular users, super users, and the application coordinator) did, and how EUD was coordinated between super users and the application coordinator. Our findings show that super users fill an important niche as mediators between regular users and local developers and can make a significant contribution to the success of EUD efforts in a non-technical application domain.

Introduction

The concept of co-worker competence has become a matter of interest in many companies in Scandinavia as well as elsewhere in the world. Work has become significantly more complex as workers confront the integration of traditional work with computer use. Employees in the modern workplace need to master new tools while continuing to employ older skills. In addition, the available task-relevant information has mushroomed. In knowledge intensive domains, such as accounting, this involves immediate access to large amounts of information (e.g. all the rules for income tax returns for various enterprises).

The development of co-worker competence must keep pace with the introduction of new technology (Ellström, Gustavsson & Larsson, 1996; Edwards, 1997). It raises the importance of how organizations employ Information and Communication Technology (ICT) to adapt the workplace to the learning needs of diverse employees, or increase the flexibility of technical support (for example, using local expertise vs. buying software for adaptation). For the company in our study, we focused our observations and interviews on the technical infrastructure surrounding a new business application. This perspective enabled us to analyze activities associated with End User Development (EUD) and learning at work, two important aspects in the adoption and use of generic, multi-purpose applications. We have chosen a framework for analysis based on Activity Theory (Engeström, 1987; Kuutti, 1996; Nardi, 1996; Kaptelinin, 1996). This framework provides explanatory categories that allowed us to focus both on the integration of work and learning and the integration of design and use. We find that these four related issues need to be addressed when introducing complex application systems into an organization with a large and diverse user group. In our case study, the support needs of the users (accountants) varied depending on the clients with whom they interact (from small and medium-sized businesses (SMBs) to large enterprises).

Previously the problem of introducing new technology has been addressed by bringing users and professional developers closer together, for example, conducting user testing in developer laboratories; developing in-house software systems, instituting company-wide

teaching programs, etc. (Grudin, 1991). The case we analyzed had a different goal, namely to bridge the gap between developers and users by creating new user/developer roles. Furthermore, these roles were acknowledged and supported by management in the organization and they persisted after the technology had been put into use.

We present a case study of "super users", who we define as *"regular employees with in-depth knowledge of one or more of the organization's computer applications without being programmers."* Super users have both domain expertise and computer know-how, and they are trained to teach other users. They are not trained as programmers; instead they interact with regular users and with local developers in their daily work. We analyzed how an organization successfully initiates a program to train super users in conjunction with introducing a new software application, Visma Business (VB). Based on empirical material, we discuss our experiences and summarize our findings. Our research was formulated to address the following questions:

- How do super users engage in EUD activities in order to achieve an efficient use of a complex computer application?
- How are EUD activities organized (roles, division of labor, etc.)?

The questions are discussed throughout the paper and specifically addressed in the empirical section. The rest of this paper is organized as follows. We start by presenting the perspective and rationale of the study, followed by a brief survey of EUD and user design activities. Next, we present our conceptual framework, based on elements of Activity Theory. Then we provide background information of the company we studied and present the case in some detail, including an overview of the VB application. Next, we present the method for data analysis. The last part provides an empirical analysis of the data we collected. At the end we summarize our findings and provide suggestions for further work.

Perspective

Knowledge transfer and learning have always been a central factor in organizational implementation of Information and Communication Technology (ICT.) When an organization chooses to implement a new application system, access to information will rapidly increase. As a consequence, few individuals are able to master all information relevant for their daily work. Instead, information becomes distributed among people and stored in databases, causing an information management problem (Gantt & Nardi, 1992; Hollan, Hutchins & Kirsh, 2000; Ackermann, Pipek & Wulf, 2003). In this paper we address this information management problem at its most dramatic, in conjunction with the introduction of a generic information system. Two central issues need to be addressed to ensure a smooth transition from development to use--the integration of working and learning, and End User Development (EUD). These issues are intertwined but in this paper we focus on EUD issues. In a separate paper, we report on our study of the integration of working and learning in conjunction with a similar adoption and use case at a different company (Mørch et al., 2004a).

End-user Development

Recent studies and system building efforts (e.g. Costabile et al., 2004; Fischer et al., 2004) have shown there is a growing need for new methods and techniques for end user development. This is motivated by increased user diversity and a tendency in software houses to produce generic, multi purpose software packages for a wide range of application domains. By generic or multi purpose, we mean systems that can be configured to different organizational needs (Bansler & Havn, 1994) and can, for example, provide different user groups with different access rights to shared objects (Stevens & Wulf, 2002).

On the other side of the development process resides a varied population of end users, with different cultural, educational, training, and employment backgrounds. They are novice and experienced computer users, ranging from the young to the mature, and they have many different abilities and disabilities (Costabile et al., 2004). Furthermore, users

in companies that interact heavily with customers operate in a variety of business contexts and scenarios that cannot always be predicted in advance. Finally, the line between "users" and "developers" becomes blurred as ICT spreads. The need for EUD in these settings is therefore fundamental in order to resolve the difference between a generic system and company-specific needs, users and application domains. On the one hand, the generic system provides functionality for everyone, but that can hamper productivity for individuals and work teams with excess information and functionality that is poorly adapted. On the other hand, the localized versions address the specific needs of individual employees and work teams, but what enhances productivity at one local office may prove less useful in other parts of the company.

End-user development, along with the related areas of human-computer interface design and software engineering, has received much attention in recent years as product developers have succeeded in embedding customization tools in commercial off-the shelf systems (COTS) and making generic applications programmable (Eisenberg, 1995; Fischer et al., 2004). These practices support the tailoring of generic systems to domain-specific needs (Mørch, 1996).

The EUD-Net "network of excellence" that has been launched as a European initiative in this area (http://giove.cnuce.cnr.it/eud-net.htm) proposes the following definition of EUD: *"End-User Development is a set of methods, activities and techniques that allow people who are non-professional software developers, at some point to create or modify a software artifact* (Costabile, et al., 2004). To provide methods, activities and techniques that bridge the gap between using and creating or modifying a software artifact is no panacea. Indeed, the gradual steps from using an application to programming it are sometimes referred to as climbing a "programming mountain" (Trigg & Moran, 1987; MacLean et al., 1990; Fischer & Girgensohn, 1990; Mørch et al., 2004b). The increase in knowledge required for an end-user developer to move one step up this mountain should be proportional to complexity of the task to be accomplished with the system, and should take into account the benefits the task can provide to end users (including the pleasure of accomplishment and the satisfaction of learning to master the computer.) Otherwise the

task should be left to professional developers. Tools such as customization forms, templates for frequently performed tasks, high-level programming languages like spreadsheet macros and scripting languages, all support EUD activities, but to varying degrees. In our study we identified a set of EUD activities that were accomplished with techniques that range from de-selecting redundant functionality to creating new functionality. These activities share characteristics with customization (Mackay, 1990) and local development (Nardi, 1995), which we briefly summarize below.

In a study at MIT in the late 1980s, Mackay (1990) studied expert users of a UNIX windowing system (X-Windows). Many of the users in that study performed activities that went beyond regular use of the system. For example they would engage in meta-level activities such as setting parameters for the position of the screen's user interface objects and storing them in a start-up file. Gant and Nardi (1992) studied users of a CAD application (AutoCAD) and a spreadsheet application (Excel) to find how well users employed built-in tailoring tools. The authors found that users who tailored their applications very often worked in groups, and in those groups the authors identified one or more users described as "local developers." These particular users were sometimes referred to as "gardeners" to emphasize their ability to "nurture" a user organization and assist regular users to use the system more effectively. This term has been used by other authors to describe similar activities in other contexts (Christiansen, 1997; Mørch, 2003; Kanstrup, 2004). A local developer, or gardener, is an expert in the professional domain who also has enough computer knowledge to tailor an application to domain-specific needs. In the context of AutoCAD, this meant adding new templates (e.g. domain-specific drawing symbols) and integrating them into the application. In the context of Excel, it meant adding computational depth to everyday operations such as "sum" and "average" by creating small scripts and macros to operate across spreadsheet cells.

Volkoff et al. (2002) introduced the term "boundary spanner" to designate individuals who can bridge the gap between domain experts who have comprehensive knowledge of organizational processes and product experts who determine how the software operates. Boundary spanners serve as "brokers" between the two groups by employing various

means to explain the practices of one community to the other (Volkoff et al., 2002). Mackay (1990) has used the term "translation" to explain a related phenomenon whereby a "gatekeeper" translates a user's problem into a technical solution. Local developers, gardeners, boundary spanners, gate keepers are all roles assumed by super users.

The super users we studied in the Company engaged in the type of EUD activities just described, but they differ from the users described above. The main difference is that the super users we studied took on a contract-based role assigned to them by the company for which they work, whereas in the cases reported above, the EUD actors emerged out of the group of regular users as someone who demonstrated proficiency in using a system and who showed an interest in helping other users learn to use it efficiently as well. In the study reported by Volkoff et al (2002), boundary spanners ended up taking on two roles even though there were few rewards for the extra role. The authors suggested the need for "system sponsors" to provide special incentives for boundary spanners in order to help them maintain credibility in both communities, but in the companies reported on this was not achieved.

User-developer Continuum

Some organizations in Scandinavia use the term "super user" in conjunction with the above activities and have started to train super users to address the information overload problem associated with the introduction of advanced application systems. The term was first used, to the best of our knowledge, to name IT staff who was asked to provide technical assistance to other employees when a new system was introduced in the organization (Kaasbøll & Øgrim, 1994). Supers users were selected to take part in this activity based on their skill in using the new system, knowledge of the application domain, and their ability to teach other employees to use the system effectively. In the case we studied, the Company defined the role of super users under a contract-based agreement with a group of accountants distributed across the organization.

EUD activities can be located along a continuum from regular use to professional development. The spectrum includes:

- *Regular users*: workers who are not interested in tailoring a system, but who want to use the system's various productivity and computational tools to accomplish their required tasks.
- *Super users*: domain-trained workers who are also skilled with computers, interested in exploring meta-tools if there is time set aside for this and who like to teach other user how to use the system. Super users are boundary spanners and translators between regular users and local developers.
- *Local developers*: domain-trained workers who have more computer skill than super users. They know more about programming and have more responsibility than super users and will often be asked to coordinate the organization's EUD activities. They communicate directly with professional developers regarding development tasks that cannot be accomplished locally. In the company we report on, one person was assigned the position of "application coordinator." We see this person as a local developer in light of the EUD activities he performs. We will refer to him by the position he holds.
- *Professional developers*: IT workers who develop a new software application or a new version of an existing application. Developers work in software houses and are trained as software engineers and/or programmers.

Socio-cultural Perspective

When the Company shifted from using several accounting applications to one core application, VB, it encountered the situation of a new system impacting an existing work process. The super user initiative that was launched as a result of this process created a subsidiary function in the organization that modified the conditions for individual and collaborative communication and problem solving. To understand the various roles and relationships that were created as a result of this intervention, we need a theoretical perspective and explanatory concepts. A socio-cultural perspective allows us to talk about how human development changes through interaction and collaboration with peers while mediated by artifacts, such as ICT (Lave & Wenger, 1991; Wertsch, 1998).

The socio-cultural perspective can help us to understand the relationship between individuals and the artifacts they use and work with and the organizational units they belong to. It implies that interactions between people are mediated by artifacts, such as the technology they use (e.g. application systems). The introduction of new technology, such as an accounting application, creates a discontinuity between how tasks used to be carried and how they will be accomplished in the future using the new technology. From a socio-cultural perspective, the concept of "artifact" is used in a broad sense but with a strong focus on its mediating function (Cole, 1996; Wertsch, 1998). Understanding artifact mediation gives us some insight into how knowledge is accumulated in the organization and how it is distributed by technology and among individuals, a process that persists over time through transformations and discontinuous developments. In our case, knowledge concerning accounting was codified in the existing system in the form of rules and procedures and specialized client-specific business solutions. The new artifact (VB) creates opportunities and constraints on accomplishing required accounting tasks. To understand the change in work as a result of the new technology, we must use a conceptual framework that provides a comprehensive set of explanatory terms for bridging technology and organization. For this purpose we have chosen Activity Theory.

Activity Theory

Activity Theory (AT) provides a conceptual framework for understanding human and work-oriented developmental processes (Kuutti, 1996). It is a powerful, descriptive tool rather than a predictive theory (Nardi, 1996). Whereas in many psychological theories, human actions are used as the unit of analysis, AT takes action a step further by claiming that it is difficult to analyze real-life situations outside a laboratory without considering the context in which actions are taken. In AT the unit of analysis is human action within its minimal meaningful context (Kuutti, 1996), which is encapsulated in the term "activity." Activity unites action and context into a conceptual whole (Engeström, 1987). We will not go into depth concerning the structure of the framework of Activity Theory here. Interested readers are encouraged to consult the sources (e.g. Engeström, 1987; Kaptelinin, 1996; Kuutti, 1996; Nardi, 1996). We give a brief overview of the relevant concepts we used in our analysis.

Studying human activity means understanding the artifact-mediated and object-oriented actions of humans as they interact and collaborate within a system, and examining how these actions relate to the transformation of the activity system over time (Engeström, 2001). Activity Theory studies both individual activity systems and the interaction of multiple systems. According to the theory, individual systems are internally unstable. Interaction becomes important because it can to lead to transformations of a system over time and further development towards more stable systems (Kuutti, 1996). All activity systems are mediated by artifacts and objects, which provide the focal points of the activities. Together with subject, rules, community and division of labor, these are the basic concepts of AT (e.g. Engeström, 1987). The objects are often referred to as "boundary objects." The boundary object concept originated with Star and Griesemer (1989). It is characterized as something that is common to more than one activity system and holds them together as well as separating them. Furthermore, a boundary object can be either a physical thing (the conventional meaning of the term object) or a pre-understanding of the thing, such as plan or an idea (e.g. adapt VB to meet the needs of the accountants). It is both a given in the situation and something towards which the activity is directed, i.e. something anticipated, projected, transformed and achieved (Engeström, 1987; Kuutti, 1996). The subject in this context is the participant in the community at the work place. The relationship between the subject and the object in an activity is mediated by the object or tool. A tool can be anything used in the process of transforming the object, whether material and mental. In our case VB is the central tool, as well as its tailoring or meta-level features. We used concepts from AT to analyze the activities of super users and a local developer in the Company.

In addition to subject, object and tool, the framework contains a fourth component, community. Community consists of all those who share or interact with the same object during a given activity. Furthermore, the relationship between subject and community is mediated by rules (e.g. tax rules, business logic), and the relationship between the community and the object is mediated by the division of labor. Division of labor is useful in analyzing how super users and the local developer share and divide tasks and how this

relationship evolves. The division of labor is a fundamental part of how work and responsibility are organized in the Company. When this can be described in terms of community and objects operating in an activity system, the explanation can be put in a meaningful context.

For our purposes, the activity of the super users and local developers in the process of adapting VB to new needs forms our unit of analysis. In particular we look at how EUD activities are mediated by artifacts such as VB, VB tailoring tools, accounting practice, and computer knowledge. Our analysis takes into account data sampled from the entire EUD community in the Company, and identifies the division of labor within the community according to what the various end user developers do. We see EUD activities as several activity systems (regular users, super users, local developer, professional programmers) in their effort of working together, and we try to relate this to the various work procedures (rules) in the workplace that constrain and enable EUD.

Case

The case we have analyzed is part of the research and developmental project "Læring på ArbeidsPlassen" (LAP, Learning and knowledge building at work). The LAP project, which started in May 2002 and ended in December 2004, involved six partners, with partial funding from the Research Council.—.two industrial partners (an oil company and the accounting firm referred to as the Company) three research units, and one national federation of service companies. The project was "user-oriented," a project category defined by the Research Council, meaning that the industry partners defined the research problems the researcher should work on, while the methods, techniques and theories to address the problems were selected or developed by the research partners in collaboration. The effect of this was a division of labor when it came to implementing changes proposed during the project. These were primarily undertaken by the industry partners based on research reports and knowledge dissemination seminars delivered by the research units.

One of the industry partners, the Company, figures in the case study we describe in this paper. The Company is an accounting consulting company that is office-based, project-driven and geographically distributed, with a long-term emphasis on competence development for its employees. To that end, it has started to include ICT in its agenda. Furthermore, the Company is interested in research and development activities that can produce useful results not only for its individual employees and its overall business goals, but also for other companies that have a similar organizational structure. The Company had decided to implement a new computer application, Visma Business, and this decision was made prior to conception of the LAP project.

The Company makes its revenues by undertaking accounting and tax consultancy services for SMBs and large enterprises. It has around 1000 employees distributed in 75 offices across Scandinavia. The Company has expanded in the last few years through the acquisition of new offices. Until recently, the Company had used a total of thirteen different accounting systems to support the work of its accountants. A decision was made in 2001 that all offices should convert to one single application, called Visma Business, or VB. This is a large and complex generic computer application consisting of several modules covering all aspects of accounting. The employees in the Company work with different clients and therefore have different requirements regarding which functionality they need or don't need. After the completion of the adoption process, VB became the main tool for all employees in the company, and for most of them, this has had an impact on the tasks they perform. In anticipation of this expected interaction between tools and tasks, all of the employees had to go through an introductory course in VB. The course combined e-learning and face-to-face meetings in a classroom. In addition to regular users learning how to use the system, super users were also selected and given additional training. They would play the dual role of teacher and local developer: to reinforce the training to colleagues in their office after the introductory course and to adapt VB to local client needs. The decision to cultivate super users was also made by the Company before they joined the LAP project. In this paper we take a closer look at the super user program, the role played by super users, the activities they carried out during adoption and use of VB, and the other employees they interacted with during EUD activities.

After completing the initial training, the super users went back to their offices with the new responsibility of helping their colleagues use the system, which in many cases meant adapting it to their needs using EUD techniques.[1] They did not rely on a central technical support unit for this purpose. The Company already had an IT department, but they chose to position super users in the zone between technical support and employees. In case of technical difficulties, the super user could contact IT for technical assistance, but this channel was rarely used. One reason for this (on which we elaborate in the empirical part) is that while the IT department provides technical support, they do not know the profession-oriented language of accounting very well, which was a key to understanding the reason for many of the EUD activities with VB.

Super Users and the Application Coordinator

The Company, as mentioned above, chose to initiate a super user program in the hope of achieving a successful adoption of the new application (Åsand et al., 2004). They decided to have one super user for every ten employees, and these were chosen from among the ordinary employees in the local offices. The corporate management formalized the criteria for being a super user in form of a contract. Both the local office manager and each super user had to sign the contract. The contract lays out the duties and expectations of the super user role, as follows:

- The super user must be competent both professionally and technically, with the emphasis on mastery of the profession-oriented accounting language.
- The super user must set aside time for training, and for sharing knowledge, for example, conducting workshops for their office colleagues.
- The super user must provide all employees in their local office with the necessary training to use the new VB application for their specific accounting needs, and to manage a specific schedule for such training. The latter requirement was of great importance and was made explicit in the contract. Without scheduled presentations, the training may not be as effective.

[1] Their previous responsibilities were adjusted accordingly

By the use of this contract, the Company formalized the super user role, which conferred legitimacy and visibility on the time and effort invested by the super users. After they signed the contract, the super users received additional, more thorough training in VB that focused on the more technical aspects of the application, especially the EUD features.

In addition to the super users, the company appointed one person to be responsible for coordinating all the super user activities and giving them training. This person, called the application coordinator, is the only appointed local developer in the Company. However, he is a trained accountant. He showed a special interest in the new application from the very beginning of the project, and was asked to manage the super users and the centralized EUD activities, in a full time position. He is also responsible for reporting problems to the software house that develops VB, including errors and adaptation tasks that could not be solved in the Company, based on feedback from the super users and their colleagues. The adaptation tasks handled by the application coordinator include adding new menus and new fields to existing applications in order to evolve them into domain-specific business solutions. The results are referred to as screens and resemble spreadsheet applications, and the method employed can be seen as an instance of evolutionary application development (Mørch, 1996). If new business rules required new functionality in the software, those modifications were handled by developers in the software house. A new solution typically starts with a request from an accountant for a change to an existing solution based on the specific needs of a client. The simpler tailoring tasks are handled locally by an office super user, but more complicated tasks are referred to the application coordinator. The application coordinator has enhanced access rights to the system, allowing him to distribute generally useful solutions to other offices as well. Through a centralized application system, a locally adapted business solution can be made “global” and therefore accessible to everyone using VB. Due to the amount of work required to create specialized solutions, the application coordinator has gradually given access rights to some of the other super users as well so that they can help him to speed up the process. These super users evolved into the role of *local developer* because

they showed more interest than the average super users in creating solutions with the tailoring tools in VB. These local developers collaborate closely with the other super users in their office, who are in a similar position as they, and with the application coordinator.

The Application - Visma Business (VB)

Visma Business is a complex accounting application, consisting of several modules, providing a superset of the functionality of the old systems. This is why we call it a generic system (Bansler & Havn, 1994). On a daily basis, much of the generic system is not used, since it includes too much information for any one accountant. The range of possible ways of doing things makes it very frustrating to learn. There are many functions and possibilities and only a few users can employ them all. Actually, 95% of the time spent in VB is spent in a few accounting modules. In addition, VB comes with a small set of business solutions for client-specific tasks available for everyone, and a set of tailoring tools (a meta-level design mode) to create new specialized business solutions, to modify existing solutions, and to remove superfluous (unneeded) functionality.

Insert Figure 1 about here: Visma Business, start window (generic system)

The business solutions are normally tailored for specific offices, depending on the clients the offices do business with. So even though VB starts as a generic application, end-users can make changes and tailor it to their use. The changes are made locally, but those found to have company-wide application can also be enabled "globally" by the application coordinator, as previously described. Possible changes include choosing which functions to make available and modifying or creating menus and fields. Much of this activity is similar to creating spreadsheets (see Figure 2). In fact VB is closely related to the spreadsheet concept and can actually be connected with Excel sheets. New functionality can be defined in Excel cells and linked to VB, making them accessible through VB screens. Access to the meta-level tailoring tools is accomplished by switching to a design mode. This provides the application coordinator and some super users with an expanded set of commands for making such changes. This mode is controlled by an administrative

password that only the local developer and super users have access to. Simpler tailoring tasks, such as modifying the name of existing field or menu is also accomplished in design mode.

Insert Figure 2 about here: Tailoring VB provides access to an expanded command set. A field "Sum" has been created and added to a business solution for a real estate agent.

The above screenshot is an example of the design mode used by the local developer when making changes in the system. For regular users without appropriate rights, these choices are not available and appear as unavailable functions (i.e. on grayed-out menus).

Method

As a basis for the analysis, we conducted interviews with a broad sample of super users as well as with other employees in the Company. The main purpose was to identify various relationships that arose between super users and their colleagues as a result of the introduction of VB, and how these relationships fit into the larger context of professional development, learning and EUD activities in the work place.

The interviews were conducted from December 2002 through May 2003, and were grouped in three rounds. In total, 16 super users and 23 regular employees were interviewed. The respondents were distributed across 9 offices and the selection criteria included gender, number of employees at a given location, geographical location and phase of converting to VB. With the selected super users as a starting point, we selected the 23 regular employees based on their contact with these super users. The first 10 interviews were carried out by the researchers during visits to local offices, while the remaining interviews were performed by telephone. Each interview started with an e-mail introduction from the management of the Company, followed by a more detailed e-mail about the procedure, sent out by the LAP-project researchers that would conduct the interviews (one Ph.D. and two master's students). Each interview took approximately 15 to 20 minutes. Confidentiality was of great importance, and all interviews were carried out by researchers and not by the Company. By focusing on the research as the agenda,

we hoped to capture the variety in experiences related to being a super user or being a regular user interacting with a super user. In addition to these interviews, we conducted interviews with the director of competence, and the application coordinator. All interviews were taped and transcribed. These transcripts together with the contract the super users signed constitute the basis for the analysis we present next.

Empirical Analysis

Activity theory provides a set of concepts that is helpful to analyze our case. The emphasis on *activity* brings certain features of a developmental process to the forefront that might otherwise be overlooked when studying interaction with technology. For example there are different actors involved in EUD (regular users, super users and application coordinator) with different goals and different artifacts to accomplish their work, regulated by different work procedures (rules). To explain how these phenomena are interrelated we have chosen to use the concepts of subjects, community, objects, artifacts and division of labor in our analysis. For example the relationship between the artifact (VB or the professional domain) and the object (the reason for EUD or customization) is reciprocal, creating a kind of task-artifact-task cycle (Carroll et al., 1992). But there are other dependency relationships as well. Subjects, artifacts and objects are affected by the existing rules, the community in which the activity occurs and the division of labor within this community. We analyze the following topics in more detail in the following subsections:

- The profession of accounting as a boundary object;
- The VB application as a meditating artifact;
- The division of labor among the different user groups and developers.

The first topic is of special importance in that it stands as a prerequisite for the other two. It concerns the process that leads up to EUD, whereas the other two shows the results after a boundary object has been successfully established.

The Profession of Accounting as a Boundary Object

Star and Griesemer (1989) introduced boundary objects as objects that are both flexible enough to adapt to local needs and constrained enough to allow several parties to employ them for their own purposes. These objects are robust enough to maintain a common identity across communities (activity systems). Boundary objects should therefore be seen as both abstract and concrete and they have different meanings in different communities. The creation and management of boundary objects thus becomes a key task in order to develop and maintain coherence across interacting activity systems.

When regular users, super users, the application coordinator (local developer) and sometimes the professional developers talk about VB together, it is important that they have a shared object in mind or in front of them to ground the discussion. This can be the application per se or the profession-oriented language of the domain. The latter provides a pre-understanding for using and further developing the application. We see both of them as boundary objects. Previous studies have shown that both the "application system language" and the profession-oriented language of the domain have to be learned for collaboration between developers and users to succeed (Nygaard, 1984).

It was the opinion among all employees interviewed that it was positive to have a super user acting as the local expert at each office. This made it possible to have a professional discussion of accounting issues with the confidence that the super user could understand the problems the employees were experiencing. It was also commonly accepted that technical competence and the ability to grasp new knowledge quickly were very important requirements for being a super user. One of the super users put it this way:

> "I think it is important to have both. We noticed it at the same time we converted to VB, when we called the support unit. Many of them managed the technical bit, but nobody had any clue about accounting. It was a problem. When the super users are here in the house they use the same applications. We talk the same language."

The employee here points to the importance of a shared understanding (boundary object) by saying that the super users need to have the same understanding of the problem as the person stating the problem. They interact within the same community of practice (Wenger, 1998) as the regular users, where the shared object constitutes the basis for the collective competence that develops. This is what, in the context of traditional system development, Nygaard refers to as "profession-oriented language" or "user-oriented languages" (Nygaard, 1984) and what, in the context of AT, Engeström (2001) calls a "shared repertoire." To develop a shared repertoire requires a reciprocal engagement regarding the task of solving problems (Wenger, 1998). When problems requiring EUD occur, an engagement is created by the involved parties in such a way that super users, the application coordinator and regular users can all learn something new. The super user's motivation and learning potential is in generalizing from the situation to solve related problems in the future. The application coordinator sees an opportunity to improve the application with a new business solution that might apply across all local offices. The regular user wishes to solve a problem for a client (e.g. completing a budget on time). Hence, they all have a wish to understand the situation.

When a shared repertoire has been established, one can then start to close the gap between the generic VB application and the requirements of a tailored version. To examine what kind of knowledge is required to accomplish this is the next stage in the process. The emphasis on a shared object is maintained, and the additional application system knowledge required is built on this basis. The bottom-up direction of competence development was stressed by all users interviewed. It starts with a foundation of knowledge that the IT support unit and the professional developers do not hold.

> "I pay bills directly for a customer and it is a sensitive operation, then the money goes directly out from the bank, and in this process I have used the super user a lot when I was insecure. The super user does the same operation for his clients and we have had a close collaboration in this part of VB, and it has been under construction, and I have therefore been very careful. It's been good to have a super user."

Security is important when providing a service for a customer, especially when one has the full responsibility for the clients' financial statements. It becomes even more important when providing a service while trying out a new accounting application. The above excerpt shows how important this security was as a reason for seeking assistance from an expert. The professional integrity required by an accountant in their interactions with colleagues and clients is something the IT staff and professional developers do not have the background to fully understand. The trust that exists among those who already know the profession is a very valuable resource when solving problems. Although developers have competence in technical solutions, they do not see the possibilities and constraints of various solutions from the user's point of view the way a super user can.

It is through their professional competence that super users and the local developer have developed their skills as translators (Mackay, 1990) and boundary spanners (Volkoff et al., 2002). At the same time it becomes important that the "translation" from a domain-specific problem to a technical solution be accomplished. What is central in the above excerpt is the fact that one has to take the client's perspective into account when solving the problem. With the super users' knowledge of the new application they can act as translators (Mackay, 1990) and boundary spanners (Volkoff, 2002) for the regular users, as, for example, when a super user assists in the process of finding the right functionality for a regular user based on their clients' needs. Furthermore, some of the super users make changes in VB to facilitate certain accounting processes by creating new menus and fields or renaming existing ones. This overview of the application is not something a regular user has.

The VB Application as a Mediating Artifact

Using VB to mediate the relationship between the subject users and the object is also a process aimed at closing the gap mentioned earlier. Adaptation of VB to domain-specific use will be the task of the super users in collaboration with the application coordinator. The accountants have various clients they keep accounts for, and to offer good service to the clients, they depend on finding the right functions in VB, making them available, and

to ensure the workflow is optimal. By learning the proper way of working in VB, the super users tailor the application for each client by selecting and deselecting functionality. To be able to modify a function such as a button or field, they may have to get assistance from someone who has expertise in the underlying functionality in VB. This activity is referred to as boundary spanning (Volkoff et al., 2002). The super users communicate with both regular users and the application coordinator, assisting the former in finding the right functions in VB, consulting the latter when the task is beyond their skill and training. One of the users, described VB as follows:

> "VB is not a completed program and may never be. It is a pile of Lego blocks that you have to put together to create an accounting system suiting the needs of the clients you keep accounts for. It's clearly a high threshold to cross."

This excerpt as viewed under an AT framework defines mediation through an artifact. When a regular user and a super user discuss how to tailor VB to achieve the best result for working with a client's account, they use VB as a mediating artifact or tool for the purpose of accomplishing the object, which is to do accounting for the client. In this tailoring process, the super user helps the regular user to choose the options that best fit the task to be accomplished. This mediation can also be seen when the super user takes a more complex problem to the application coordinator. Together they discuss possible changes to be made to the application, and sometimes they end up with a new business solution tailored to a specific client or a branch of clients.

> "The changes we make are often tailor-made solutions for a particular group of clients, such as real-estate agents. When a group of clients is relatively large we produce shared solutions for them, which make doing their accounts more streamlined. The request for these kinds of solutions comes from those who do the accounts."

It is not always the super users who discover the solution to a problem. Often a regular user, through their position in the organization, is able to bring the solution to the

attention of other VB users. This is achieved as a result of collaboration between local developers, super users and regular users. This is also something Gantt and Nardi (1992) have explored; that user and local developers often collaborate to make adjustments that one person in the group cannot do on his own.

Insert Figure 3 about here: A business solution for real estate agents

Figure 3 shows the development of a business solution for real estate agents. The application coordinator and a super user have developed this solution by tailoring VB. They have removed standard fields and added new ones to model their clients' accounts better. All fields in VB can be hidden and made visible again. They can also be sorted and selected.

Much of the tailoring is done in design mode and the changes made in this mode can very easily be made global for all users of VB, which is a feature of VB's centralized architecture. In the beginning of the implementation process of VB, the management and the application coordinator discussed how wide the access should be. They agreed to start with an open approach, by which all super users had access to the design mode and could thus save their changes globally in VB. After a period of testing the access rights became restricted, and only the local developer and some chosen super users were allowed this access right. Having the access rights available to all super users became a problem because not every super user had the need to make their solutions global. In fact, when this feature was overused it tended to be disruptive to everyday work as regular users were overloaded with functionality they did not need. As it is now implemented, it works well. However, this still depends on good communication between those super users who do not have full access rights and those who do, such as the application coordinator.

The Division of Labor between the Different User Groups and Developers

We see that the various roles of regular user, super user, local developer and professional developer differ in responsibilities regarding EUD activities. A typical difference is the access rights available when making changes global as we just described. Another

difference related to making the local changes. Regular users have the option to select only those parts of the application they want to use, and the super users help them in doing this because they have a better overview of the VB functionality and what can and cannot be done when making such modifications. One super user put it this way when asked about the division of labor between him and the others users in his office:

> "Regular users do not make changes in VB, I do it. Users notify me about needs for making changes or modifying modules in VB. Some of the super users also have this access. "

Regular users cannot make changes and save them. These tasks are performed by the local developer and by some of the super users. There is an explicit organization to the community of users regarding the transformation of the object into outcome. The object is here to make VB work in interactions with their clients, and to achieve this they need to cooperate within their community to make the changes in VB. A similar division of labor can be found between the application coordinator and the software house, in situations where the local developer has to consult the developers to solve a EUD problem that involves application programming. The following comment from the application coordinator illustrates the situation:

> "I don't do any programming within VB (Visma Business). For that I need the professional developers. I report possible improvements, and maybe we see a result of it in the next version."

The EUD activities performed by the application coordinator include selecting modules to be part of a new business solution and modifying the menus and fields within new and existing solutions. The local developers cannot modify the program behind the menus and fields because this is written in a proprietary language. Programming of this kind is judged to be too complex for someone not trained as a programmer and is therefore left to professional developers in the software house. It is more complex than writing macros in

Excel, which is a task most super users and even some regular users are quite familiar with, and sometimes workarounds are made by integrating VB with Excel functionality.

> ”As application coordinator, I have access to make and save changes to VB. These possibilities are not available to the regular users; it is only a few selected super users and me. This difference is made visible in VB in that the tailoring tools are “grayed out” in the regular users’ interface.”

Technologically the division of labor between local developers and regular users is manifested by the two levels of access rights to the system’s tailoring features (see Figure 2). In this way, division of labor enables boundary objects to serve as a link between the community of users and the community of developers. The Company experimented for a while before they decided to modify the access rights. This may be a result of the size of the company (close to 1000 accountants) and the multitude of specialized solutions that were created with VB. To make the system manageable, it became necessary to restrict full tailoring rights to only a handful of super users.

Summary and Conclusions

We have presented a case study of the organizational implementation of a complex business application in an accounting Company. During this process, the Company chose to use super users and a local developer, referred to as application coordinator, to support the adoption and use of the system, including tailoring it to the needs of local offices. We observed the EUD activities and collected data by interviewing regular users, super users, and the application coordinator. In the analysis we identified the importance of the professional language of accounting as a boundary object, as well as the VB application as a mediating artifact. The primary result of these activities is a tailored set of VB solutions. These local solutions are created with the help of super users and the application coordinator in collaboration, but the need for solutions is brought to their attention by the regular users (accountants).

By taking a socio-cultural approach, we have focused on the overall organization of EUD activity, and by applying concepts from Activity Theory; we are able to explain a set of interrelated phenomenon. The object is the not-quite-ready generic application, which is transformed into a user-oriented application supporting accountants in their interactions with clients and with the domain knowledge held by the experienced accountants. The regular users, the super users and the application coordinator are the community that shares these objects, occasionally joined by one or more professional developers. There is a certain division of labor within the community, 1) between the regular users and the super users, 2) between the super users and the application coordinator, and 3) between the application coordinator and the professional developers. There is also a set of rules, defined by tax laws and the management in the Company, or implicit by a general working culture established in the Company as the users work together. The business logic programmed in VB functionality also provides rules at a more detailed level. In the application transformation process, various tools are used. For example both the application system language (e.g. business logic) and the professional language of the domain (accounting) interact mediated by the VB tools during the activity.

Our main findings from the analysis are three criteria we believe important for an organization desiring the successful implementation of a complex computer application:

- *Institutionalization of super user initiative:* EUD became institutionalized in the Company with the strategic decision to involve super users in the implementation process. This was in response to a complex and multi-purpose system (VB). The role of super users was established through a contract, which guaranteed that the super users had time to perform the activities the role required.
- *Grounding*: The Company chose to have a distributed network of super users located at all of its offices (one super user for every ten accountants). The grounding was accomplished in two ways: 1) geographical distribution of super users, and 2) utilization of super users with a background in the same profession as the regular users.

- *Local developer as coordinator*: The Company chose to have one person holding the position as an application coordinator. This person's responsibility is primarily to perform EUD activities at a general level and work closely with some of the more experienced super users in the offices as well as communicating with the professional developers. This person generalizes the results of useful local EUD activities and makes them available throughout the Company. This organization of responsibility enabled a smooth transition from everyday use and user requirements capture to development of new functionality because the EUD activity is grounded in the regular users' daily work.

References

Ackerman, M.S., Pipek, V. & Wulf, V. (2003). *Sharing Expertise: Beyond Knowledge Management*. The MIT Press: Cambridge, MA.

Bansler, J.P. & Havn, E. (1994). Information Systems Development with Generic Systems. In Walter R.J. Baets (Ed.), *Proceedings from Second European Conference on Information Systems*. Nijenrode University (pp. 30 – 31) Breukelen: Nijenrode University Press.

Carroll, J. M., Kellogg, W. A. & Rosson, M. B. (1992). The Task-Artifact Cycle. In Carroll, J. M. (Ed.) *Designing Interaction: psychology at the human-computer interface*. (pp. 74-102).Cambridge Series on Human-Computer Interaction.

Christiansen, E. (1997). Gardening: A Metaphor for Sustainability in Information Technology-Technical Support. In J. Berleur & D. Whitehouse (Eds.), *An Ethical Global Information Society: Culture and Democracy Revisited*. London: Chapman & Hall.

Cole, M. (1996). *Cultural Psychology. A once and future discipline*. Cambridge Mass.: Harvard University Press.

Costabile, M.F., Foglia, D., Fresta, G., Mussio, P. and Piccinno, A. (2004). Software Environments for End-User Development and Tailoring. *PsychNology Journal, Vol. 2, No. 1,* pp. 99-122.

Edwards, R. (1997). *Changing places : Flexibility, lifelong learning, and a learning society*. London: Routledge.

Eisenberg, M. (1995). Programmable Applications: Interpreter Meets Interface. *SIGCHI Bulletin, 27*(2), 68-83.

Ellström, P. E. , Gustavsson, B. & Larsson, S. (1996). *Livslångt lärande*. Lund: Studentlitteratur.

Engeström, Y. (1987). *Learning by expanding: An activity- theoretical approach to developmental research*. Helsinki: Orienta-Konsultit.

Engeström, Y. (2001). Expansive Learning at Work: Towards an Activity Theoretical Reconceptualization. *Journal of Education and Work 14*, 133-156.

Fischer, G. & Girgensohn, A. (1990). *End-user modifiability in design environments*, Proceedings of the Conference on Human Factors in Computing Systems (CHI'90), 183-192. New York: ACM Press.

Fischer, G., Giaccardi, E. Ye, Y., Sutcliffe, A.G. & Mehandjiev, N. (2004). Meta-design: a manifesto for end-user development. *Commun. ACM 47*(9), 33-37.

Gantt, M. & Nardi, B. (1992). *Gardeners and Gurus: Patterns of Cooperation among CAD Users*. Proceedings of the Conference on Computer-Human Interaction (CHI '92), 107-117. New York: ACM Press.

Grudin, J. (1991). Interactive Systems: Bridging the Gaps Between Developers and Users. IEEE Computer 24 (4), 59-69.

Hollan, J., Hutchins, E. & Kirsh, D. (2000). Distributed Cognition: Toward a New Foundation for Human-Computer Interaction Research. *ACM Transactions on Computer-Human Interaction 7* (2), 174-196.

Kanstrup. A-M. (2004). E-learning Behind the Facade: The Value of Local Gardeners. In E-learning at Work, edited by Kanstrup, A.M. Roskilde Universitesforlag, DK, pp- 149-166 (in Danish).

Kaptelinin, V. (1996). Activity theory: implications for human-computer interaction. In Nardi, B. (Ed), *Context and Consciousness: Activity theory and human-computer interaction* (pp. 103-116) Cambridge: MIT Press.

Kuutti, K. (1996). Activity Theory as a potential framework for Human-Computer Interaction Research. In Nardi, B. (Ed), *Context and Consciousness: Activity theory and human-computer interaction* (pp. 17-44). Cambridge: MIT Press.

Kaasbøll, J. & Øgrim, L. (1994).*Super-Users: Hackers, Management Hostages or Working Class Heroes? A Study of User Influence on Redesign in Distributed Organizations*. Proceedings of the 17th Information Systems Research Seminar in Scandinavia (IRIS-17), 784-798. Dept. of Information Processing Science, University of Oulu.

Lave, J. & E. Wenger. (1991). *Situated learning: Legitimate Peripheral Participation*. Cambridge: University Press.

Mackay, W. E. (1990). *Patterns of Sharing Customizable Software*. Proceedings of Conference on Computer Supported Cooperative Work (CSCW'90), 209-221.New York: ACM Press.

MacLean, A., Carter, K., Lövstrand, L. & Moran, T. (1990). *User-Tailorable Systems: Pressing the Issue with Buttons*. Proceedings of the Conference on Human Factors in Computing Systems (CHI'90), 175-182. ACM Press, New York.

Mørch, A. (1996). Evolving a Generic Application into a Domain-Oriented Design Environment. *Scandinavian Journal of Information Systems, 8*, (2) 63-90.

Mørch, A.I. (2003). Evolutionary Growth and Control in User Tailorable Systems. In N. Patel (Ed.), *Adaptive Evolutionary Information Systems* (pp. 30-58). Idea Group Publishing.

Mørch, A.I., Engen, B.K. and Åsand, H-R. H. (2004a). The workplace as a learning laboratory: the winding road to E-learning in a Norwegian service company. Proceedings PDC 2004: 142-151.

Mørch, A.I., Stevens, G., Won, M., Klann, M., Dittrich, Y. & Wulf, V. (2004b). Component-based technologies for end-user development. *Commun. ACM 47*(9), 59-62.

Nardi, B. (1996). Studying Context: A Comparison of Activity Theory, Situated Action Models, and Distributed Cognition. In Nardi, B. (Ed), *Context and Consciousness: Activity theory and human-computer interaction* (pp. 69-102). Cambridge: MIT Press.

Nygaard, K. (1984). *User-oriented Languages*. Proceeding of Medical Informatics Europe 84 (Brussels, 1984), 38-44. Berlin: Springer-Verlag.

Pipek, V., Hinrichs, J. & Wulf, V. (2003). Sharing Expertise: Challenges for Technical Support. In Ackerman, M., Pipek, V. & Wulf, V. (Eds.), *Sharing Expertise: Beyond Knowledge Management*. (pp.111-136). Cambridge, MA: MIT Press.

Star , S. L. & Griesemer, J. R. (1989). Institutional Ecology, "Translations" and Boundary Objects: Amateurs and Professionals in Berkley's Museum of Vertebrate Zoology1907 – 39. *Social Studies of Science* 19, (3) 387 – 420.

Stevens, G. & Wulf, V. (2002). *A new dimension in access control: studying maintenance engineering across organizational boundaries*. CSCW proceedings, 196-205. New York: ACM Press.

Trigg, R.H., Moran, T.P. & Halasz, F.G. (1987). Adaptability and Tailorability. In H.-J. Bullinger, B. Shackel (Eds.), *NoteCards*. Proceedings of lNTERACT'87, 723-728. North-Holland, Amsterdam.

Volkoff, O., Strong, D. M., & Elmes, M.B. (2002). *Between a Rock and a Hard Place: Boundary Spanners in an ERP Implementation*. Proceedings of the Americas Conference on Information Systems, August 9-11, 2002, Dallas, TX, 958-962.

Wenger, E. (1998). *Communities of practice: learning, Meaning and Identity*. New York: Cambridge University Press.

Wertsch, J. (1998). *Mind as Action.* New York: Oxford University Press.

Åsand, H-R. H., Mørch, A., and Ludvigsen, S. (2004). Super users: A strategy for ICT-introduction. In E-learning at Work, edited by Kanstrup, A.M. Roskilde Universitesforlag, DK, pp- 131-147 (in Danish).

Figures

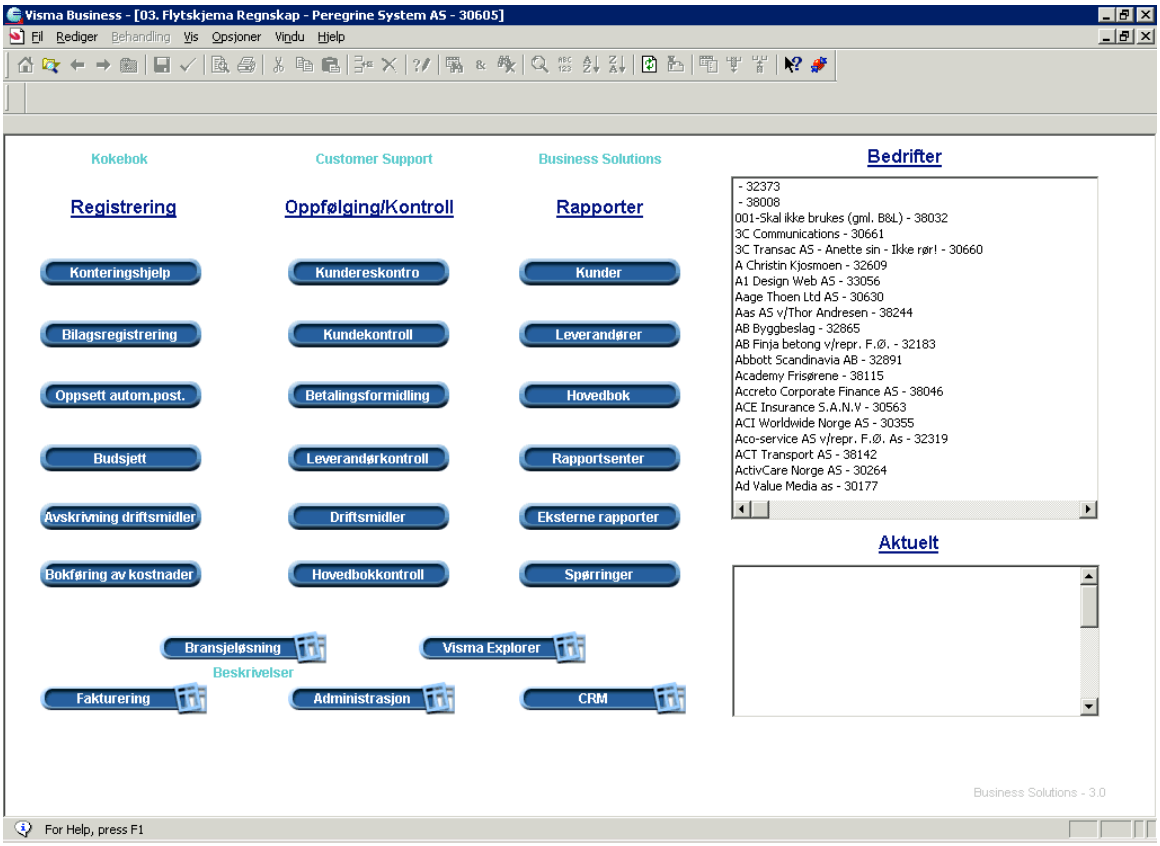

Figure 1: Visma Business, start window (generic system)

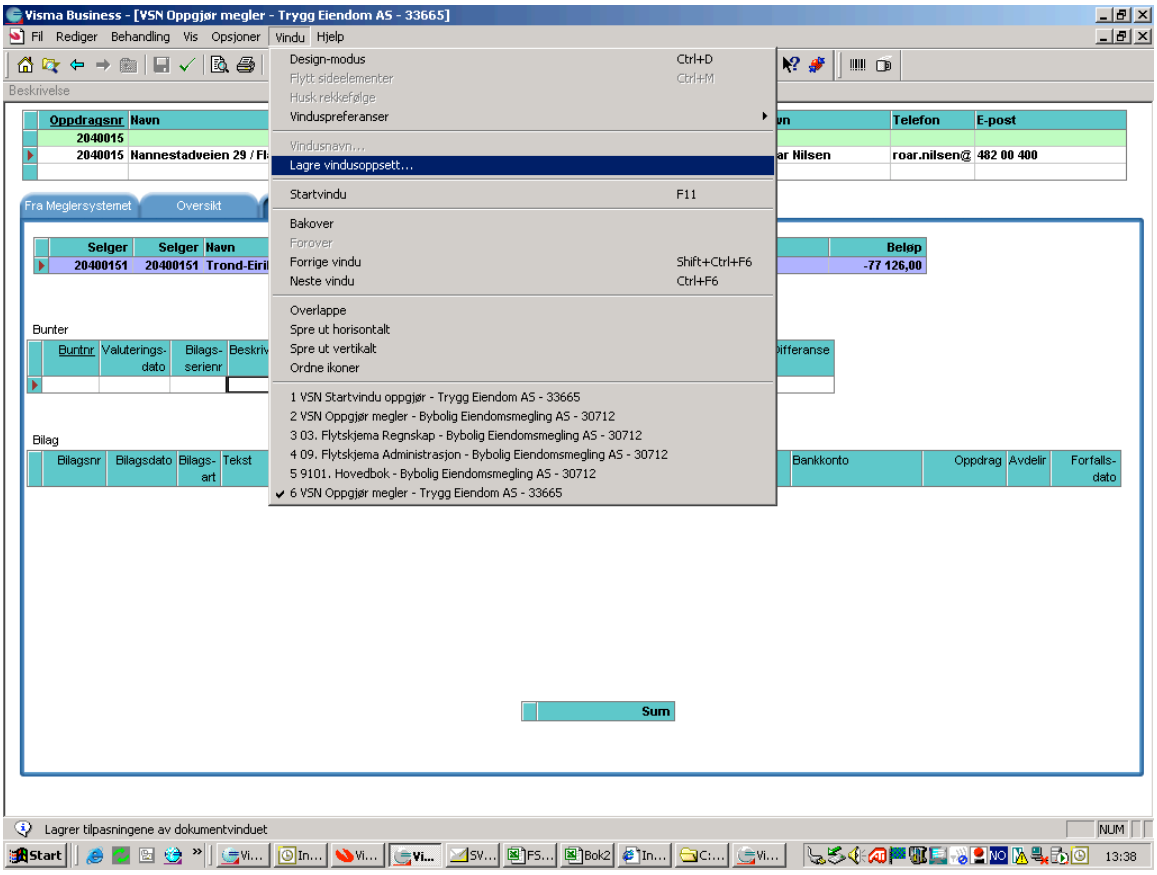

Figure 2: Tailoring VB provides access to an expanded command set. A field “Sum” has been created and added to a business solution for a real estate agent

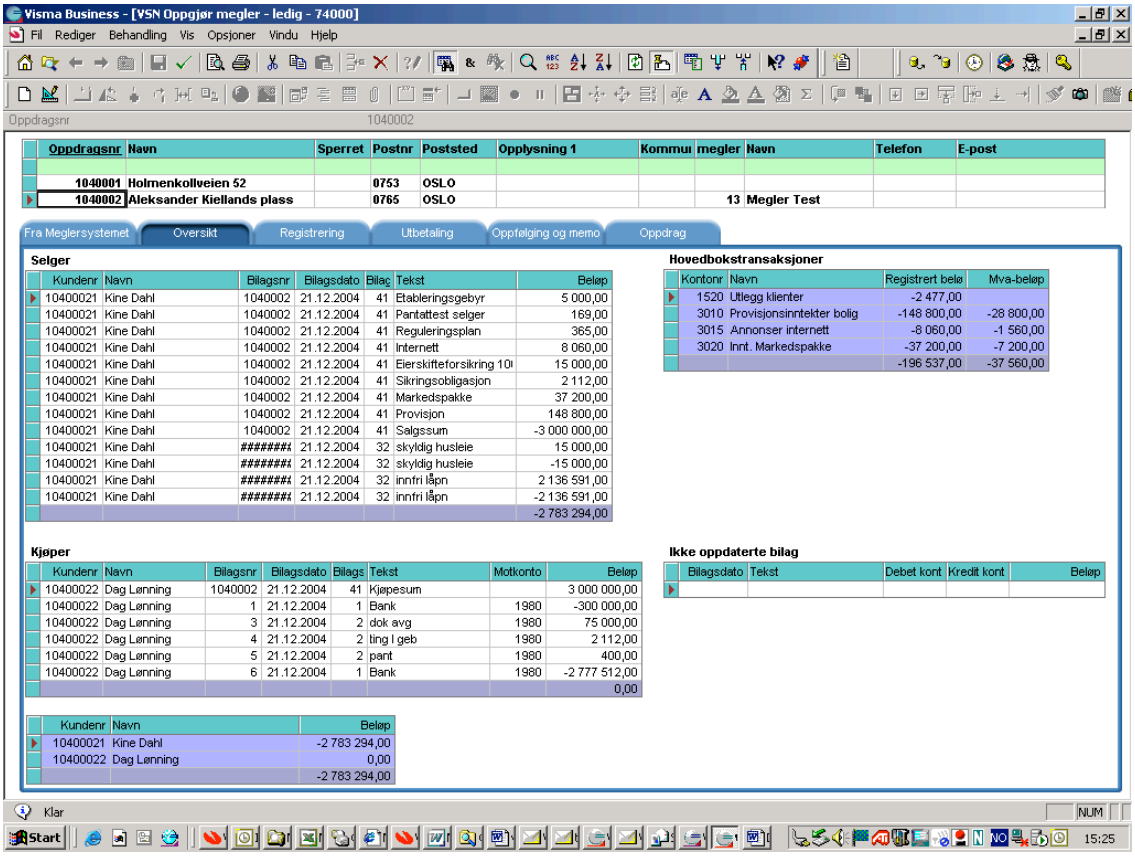

Figure 3: A business solution for real estate agents