Journal of Organizational and End User Computing
Volume 30 • Issue 1 • January-March 2018

Second Order Interactive End User Development Appropriation in the Public Sector:

Application Development Using Spreadsheet Programs

Anders Avdic, Dalarna University, Falun, Sweden

ABSTRACT

This paper seeks to respond to the research question: How does appropriation take place in the public sector in the development of end user applications by civil servants? Appropriation is defined as taking advantage of opportunities related to the development and use of applications, when the developer has in-depth knowledge of the problem domain and is also the primary user of the application. The author's results showed that public servants who have deep problem domain knowledge can take advantage of end user tools (e.g. spreadsheet programs) in the problem- solving process to solve vaguely defined problems. Appropriation is manifested in the continuous development of various ICT applications. In this paper, the author differentiates between first- and second-order appropriation. First-order appropriation takes place when the potential of the development tool is appropriated by the end user. Second-order appropriation takes place when an application is continuously developed and refined in parallel with the end user's learning process and the development of organizational requirements.

KEYWORDS

Appropriation, E-Government, End User Development, EUD, Public Sector, Spreadsheet

1. INTRODUCTION

Information systems are crucial to any organization, private as well as public. Traditionally, information systems are developed by experts within an ICT department or from some external consultancy. End User Development (EUD) implies the decentralization of systems development in an organization, rather than centralized development managed by an ICT department. Instead of the ICT department organizing systems development, individuals at various organizational levels can, within given constraints, decide for themselves when and how to develop a system meant to support the individual in his/her work. EUD takes place in private and public organizations. The use of ICT in the public sector is called eGovernment. The World Bank (2011) defines eGovernment as "the use by government agencies of information technologies that have the ability to transform relations with citizens, businesses, and other arms of government". The public sector differs from the private sector with respect to how it is owned (mostly by citizens), funded (mostly by tax), and controlled (mostly by political forces) (Perry and Rainy, 1988). These aspects of EUD in the public sector give rise to conditions that are different to those found in the private sector. Civil servants in a municipality are

DOI: 10.4018/JOEUC.2018010105

accountable to the citizens of that municipality. Consequently, they have to adjust their deeds and applications to conform to the policies of the current political majority.

EUD can be defined as "… a set of methods, techniques, and tools that allow users of software systems, who are acting as non-professional software developers, at some point to create, modify or extend a software artefact" (Lieberman et al., 2006). This implies that workers in an organization develop ICT applications for their own needs. These end users are experts in their own specific part of the organization. They use all sorts of tools and applications to carry out their work, including spreadsheet programs, CAD tools and database tools. ICT tools may be used because of a need for decision support, to double check and search for new procedures, to avoid backlogs, and to rationalize or simply explore and develop an end user's role in an organization (Fahy & Murphy, 1999; Avdic, 2009; Pankowska, 2011). Whilst any kind of ICT tool can be used in EUD, the most common is the spreadsheet program and the most commonly used spreadsheet program is MS Excel. Exact numbers are hard to find, but in Sweden where this study took place, MS Excel is completely dominant. In the USA alone there were 13 million spreadsheet end users in 2005 (Scaffidi et al., 2005). The number of MS Excel users around the world is huge. Firman (2015) has claimed that it could be as many as 750 million.

Above, we have identified some of the reasons why EUD occurs. Alternatively, EUD may be necessary because of problems that arise when using existing programs and systems. Most organizations today offer ERP systems to their employees. Some of these systems are huge, rigid, complex and perceived as being less usable than other systems by their users. Indeed, some users hesitate to adopt them, choosing instead to develop "feral" applications. This is the case even when the ERP includes modules that would fulfil the users' needs (Houghton & Kerr, 2006). Users may choose not to use a central system such as ERP (and eventually EUD) for organizational, cultural or social reasons, such as inconsistent procurement policies. Buonanno (2005) stated that business complexity and organizational structure could prompt end users to choose EUD instead of an ERP system (Buonanno, 2005).

To date, research into the use of EUD in the public sector has mainly dealt with end users, rather than EUD, itself. Searches of research databases, including Summon, ACM Digital Library and Google Scholar, using different combinations of terms such as e-government/electronic government/digital government AND end user development/end user computing/end user software engineering, have elicited hits that are about end user participation (e.g. Karlsson et al., 2012), end user satisfaction (e.g. Goel et al., 2014), suggested frameworks (e.g. Meyer et al., 2012), or web publishing (e.g. Fogli et al., 2010). A prominent eGovernment journal like Government Information Quarterly publishes papers on end user satisfaction, usability and end user acceptance. However, the closest it has come to discussing EUD is to look at user participation. This implies a need to research and understand EUD in the public sector.

The existence of ICT departments in the public sector has not been brought into question; rather, the issue is whether or not EUD should be allowed. Centralization does have some disadvantages. In particular, the centralization of application development tends to formalize the development process, which can lead to greater inertia and inflexibility than would be the case for EUD. According to Pankowska, "Drawbacks of centralized ICT resources management result in the separation of ICT functions from the business, increase of back-up costs and lack of appropriate recognition of the end user requirements" (Pankowska, 2011, p 225). A complex ICT environment generally means that the development process is also complex. Thus, ICT specialists are needed to develop and maintain applications. Occasionally, ICT specialists are also business experts, although this is not often the case. ICT staff tend to observe the organization through the lens of ICT development rather than from the perspective of users or employers/decision takers. In particular, "…ICT-driven research as well as ICT-driven governmental change often lead governments to divert their attention to technologies per se rather than to the ways in which these tools can assist them to carry out their main public

purposes" (Dawes, 2009, p 260). When staff do not use central ERP systems, for whatever reason, they may turn to workarounds, sometimes called "feral systems" (Kerr, 2014).

We conclude that there is a need for EUD in organizations. Likewise, there is a need for domain experts to use ICT to carry out their work. Obviously, there are risks associated with staff developing their own applications. For example, risks with spreadsheet applications are related to poor documentation, personal dependence, complex solutions and security. Since there are hundreds of millions of spreadsheet users, these risks should be taken seriously. One example of this is an annual conference known as EuSpRIG (2015a), which is dedicated to looking at the risks of EUD and research into spreadsheets (see, for example, the work of Raymond Panko (Spreadsheet Errors, 2010)).

So, when we summarize the pros and cons of traditional application development and EUD we note that the number of user developers is not decreasing and that there are pros and cons to both kinds of development. This means that EUD is here to stay; thus, we need to gain a better understanding of what is happening when a domain expert uses an ICT tool to carry out his professional tasks. In this paper, we explore the concept of appropriation as a means to better understand the nature of EUD.

Appropriation is a concept used in several disciplines, including sociology, art, economy and education. From a linguistic perspective, appropriation is exercised when someone adapts to something and makes it part of their own repertoire of actions: they may, for example, use it to address work-related tasks. In Information Systems (IS), appropriation is applied by the use of software that has been developed by software designers for users. Appropriation takes place when the user uses the application in a way that has not been anticipated by the designer (Wulf & Jarke, 2005). In our paper, we focus on how people's general use of software such as spreadsheet programs addresses their information needs in their own working contexts. Our aim is to elicit the synergetic effect that can appear when users, acting in a changing organizational environment, use development tools to develop and improve their own working context.

Appropriation is here defined as: *taking advantage of opportunities related to the development and use of applications, when the developer has in-depth knowledge of the problem domain and is also the primary user of the application.*

This paper seeks to answer the following research question: *How does appropriation take place in the public sector in the development of end user applications by civil servants?*

The paper is organized as follows. After the introduction, we present and discuss the central concepts in the theory section. The method section then presents our results and analysis. Finally, we discuss the results and draw any conclusions.

2. THEORY

In this section we present and discuss the meaning of EUD and appropriation concepts in order to develop a framework for empirical study.

2.1. End User Development

End User Development (or End User Computing) came into being in the 1980s, together with some relatively (at least, for that time) user-friendly tools. Actually, most programs today are "written not by professional software developers, but by people with expertise in other domains working toward goals for which they need computational support" (Ko et al., 2011:21). The most common EUD tool is the spreadsheet program. Its unrestricted nature makes it fairly easy to use and to develop applications. This ease of use is both seen as a benefit and a disadvantage. Users can benefit by being able to provide their own ICT support without being dependent on the ICT department or external consultants. However, the fact that users can develop their own systems poses a challenge for CIOs, both in the private and the public sector; for example, many organizations warn against using spreadsheets. Nonetheless, there are about half a billion MS Excel users according to Microsoft (2013). The risks of using spreadsheets in organizations mainly relate to data integrity, security and maintenance

(Panko & Aurigemma, 2010; Powell et al., 2009). No one would argue that the widespread use of spreadsheets could be disastrous if the risks were not taken into consideration. There are many stories about spreadsheet errors resulting in small and large problems in organizations (EuSpRiG, 2015b). Several measures, such as education, auditing or dedicated methods and frameworks (Panko, 2005), have been put forward to prevent such risks. Almost all spreadsheet applications have built-in errors.

EUD does not just refer to spreadsheet programs; all kinds of software can be used, including programming languages. The most specific aspect of EUD is that the user-developer has an in-depth knowledge of the domain in which the application is developed (Avdic, 2009). The user-developer also has the necessary knowledge of how software operates, along with the tasks it must perform, the business rules it must enforce, and the validation checks it must perform (Lepouras et al., 2007). Thus, the development process is driven by needs. The user-developer can use any tool to accomplish whatever goals he/she has set. A needs-driven process means that the user develops applications for his/her own needs when there is no standard system available. The user-developer must have a certain degree of independence in his/her job in order to carry out EUD. EUD can provide decision support by calculating, analyzing or questioning something related to a work-specific task that is relevant in an organizational context where the user-developer has some domain expertise (Avdic, 2009; McGill, 2004).

One aspect that makes a spreadsheet program different from other development tools (e.g. programming languages), is its ease-of-use as a personal tool. Fahy and Murphy (1999) have described how the development of various spreadsheet applications by managers helps them to process data and refine the basis for decisions in their work. According to them, managers can take a tacit dimension into consideration when developing their own systems: "The process of systems development helped the managers verbalize and amplify the understanding of the situations they faced" (Fahy & Murphy, 1999). This aspect distinguishes EUD from expert-driven development. The fact that domain experts develop systems makes it possible to take tacit knowledge into consideration. In traditional systems development, the domain experts must describe the requirements to the developer, thereby making the knowledge explicit. Michael Polanyi, who coined the concept "tacit knowledge", stated that "we know more than we can tell" (Polanyi, 1967) and that all knowledge is actually based on tacit knowledge.

EUD applications can be developed with a particular aim in mind, regardless of existing central systems; for example, civil servants may develop EUD applications in order to calculate, simulate, present or analyze something. This application could also be a "workaround" or even a "feral" system (Kerr, 2014). Workarounds may occur for many reasons, most of them rather negative (Alter, 2014). Still, the existence of a workaround can be perceived as a symptom of something worth noting, a deviation that could lead to insights about the ICT infrastructure and how it functions. Alter stated that technology workarounds may be related to appropriation as well as affordance (Alter, 2014).

2.2. Appropriation

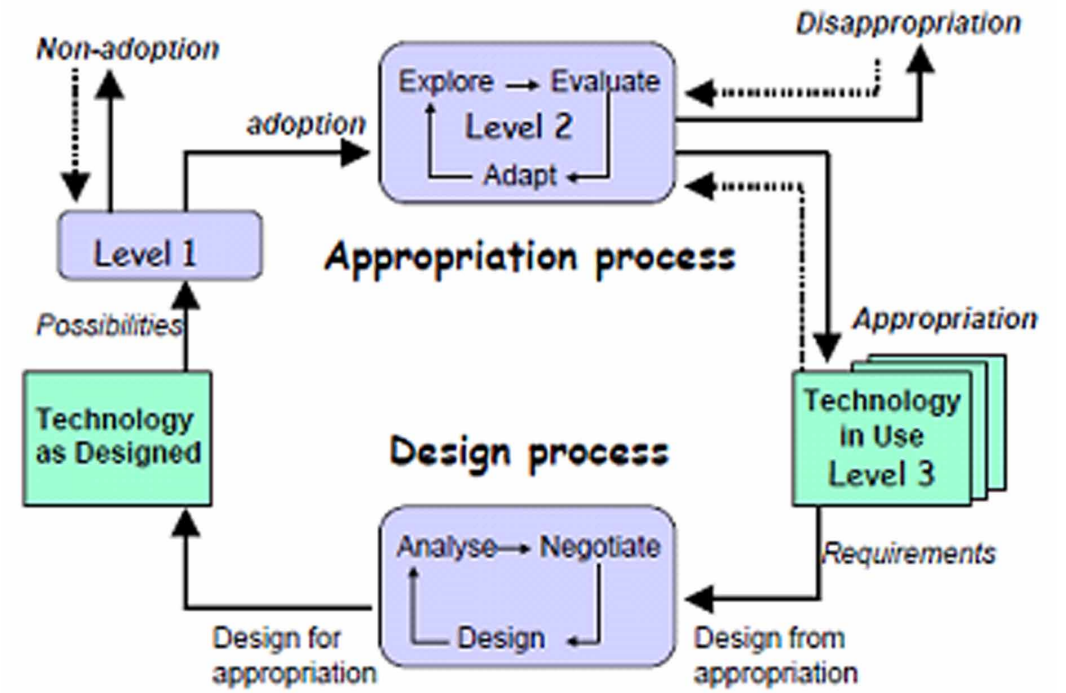
The term appropriation is applied in various domains, including law, education, socio-psychology and culture. In the field of IS, for example, it is discussed in terms of economic aspects (Wulf & Jarke, 2004) and Human Computer Interaction (HCI) (Salovaara, 2011). Similar terms, such as adoption and acceptance, are frequently used to explain or discuss problems and opportunities that relate to the use of ICT-based artifacts (Venkatesh et al., 2003; Gebauer & Ginsburg, 2009). Appropriation is also closely related to learning. It is an ongoing process and takes place when you learn how to master an intellectual or a physical tool (Säljö, 2000). When an end user uses a tool (e.g. a spreadsheet program) to develop an application that is to be used in his/her working context, there is often an element of learning involved. In the main, learning relates directly to the working context; however, it can also relate to the tool itself. Indeed, one main reason for using ICT in an end user context is to learn and question certain aspects of the end user's own working context. An end-user-developed application can help to answer questions about "how many?" or "how much?", as well as "what is the most appropriate goal?" or "how can we distribute the work load?" EUD applications can be more

or less complex. The learning activity driven by EUD does not end with a specific answer; more likely it is that of an ongoing appropriation-like process, because the working context is continuously changing. In turn, new questions are raised, giving rise to new needs for adjusting goals and means (Avdic, 2009). This is what happens when a tool or an application is appropriated. It serves as means to understanding a problem or a domain in order to explore opportunities or solve problems.

The process of technology appropriation has been described by Carroll (2003) as a cycle with three levels. According to Carroll (2003), appropriation is actually a part of the design process. Users use systems and appropriate them by "taking possession of a technology innovation over time" (Carroll, 2003, p 6). The appropriation process of a technology innovation takes place on three levels. The first level, Technology as Designed, is where a technological innovation is used in line with the designer's intentions. In this situation, the technology is adopted. On the second level, users explore, evaluate and adapt. If this is successful, users will appropriate the technology. They then enter the third level, Technology in Use, where users eventually exceed the intentions of the designer (see Figure 1 below). The output from appropriation can be expressed as the requirements for further design of the technology. It can therefore be considered as part of a continuous design process.

In her work, Carroll (2003) mainly referred to applications developed by ICT experts where the appropriated use of the application includes aspects not planned by the developer. In the case of the appropriation of EUD for spreadsheet programs, we refer to the use of an ICT tool intended for the design of applications, as well as the use of the developed applications, themselves. These two aspects are similar, but also rather different. The aim of such a tool is to develop applications. Drawing on work by Jones and Twidale (2009), the spreadsheet program can be seen as a tool that has a high degree of abstraction, which thus makes it appropriable, whereas a more dedicated and less abstract tool provides fewer opportunities for use in other contexts. Thus, information systems have an inherent potential for appropriation, we call it appropriability.

Figure 1. The technology appropriation cycle (Carroll, 2003)



Appropriation is different to and more fundamental than just configuring a standard package system in order to make it fit individual users' needs. Appropriation is a dialectic activity because it "…involves mutual adaptation: users reshape the features of an ICT, they may use it for unanticipated purposes and at the same time their practices are shaped by the ICT" (Carroll, 2003, p 2). The technology that is appropriated is an application that has the potential to be used in more than one way. For example, it could be a system that has been tailor made for a specific purpose, or it could be a standard package meant for a more general purpose. The system's degree of abstraction makes it more or less appropriable (Jones & Twidale, 2005). In this sense, a spreadsheet program is a highly appropriable tool, because it is less restricted than most other tools, and thereby can be used for a huge range of purposes.

Usage is individual and the appropriation of technology is "theoretically equivalent to psychological ownership, and thus, likely to have similar antecedents and consequences" (Gaskin & Lyytinen, 2010, p 7). Thus, the outcome of the appropriation processes in similar domains with similar applications or tools might be different from one user to another. Tools are used to mediate human activity, thereby forming an integral part of human learning and development. In this sense appropriation is closely related to the notion of affordance (Gibson, 1977; Faraj & Azad, 2012). Affordance means possibilities for action (Gibson, 1977). A tool or a system has capabilities and users can perceive different affordances depending on the users' needs or preferences (Leonardi, 2013). Users appropriate tools or systems through their affordances (Carlo et al., 2012). The more affordances a user perceives in a certain context the more appropriable is the system. When comparing a spreadsheet program with, for example, an ATM machine, the ATM machine software can be used for the withdrawal of cash or for printing an account statement, while the spreadsheet program can be used for developing an infinite number of various kinds of applications. The appropriability of ATM machine software is close to zero, whereas the appropriability of spreadsheet program capabilities makes it possible to perceive all kinds of affordances.

Leonardi (2011) used the concept of affordance when discussing human and material agency. While human agency is defined as the ability to form and realize one's goals (Giddens, 1984) material agency is "the capacity for nonhumans to act on their own, apart from human intervention" (Leonardi, 2011:147). Leonardi used the term 'nonhumans' to refer to technology; for example, spreadsheet programs. According to Leonardi, technology either constrains people's ability to achieve goals or affords the possibility of achieving new goals. The former (i.e., technology constrains) leads people to change technologies in order to accomplish goals, while the latter (i.e., technology affords) leads people to change routines. With regard to the three cases described in this paper, technology in the form of a spreadsheet program has obviously not restricted the user developers; rather, it has supported their ambition to achieve or even develop goals.

The fact that the appropriation process is individual also means that when new tools are appropriated, new opportunities to affect and be affected are attained. Appropriation always takes place in terms of concrete actions, rather than a general exposure to the tool (Berthelsen & Zander, 2006). This implies a need to specify situations in which applications are appropriated. The connection to EUD can serve this purpose.

Appropriation can be compared with the concept of "adoption", which is often used in relation to how systems are used. However, an appropriation perspective views an application not as a black box, but as a point of departure for the continuous redesigning of a process of interpenetration. In other words, the user and the application affect each other. An adoption perspective, or even an acceptance perspective, treats the application as a closed unit (Pipek, 2005; Oliveira & Martins, 2011; Karhanna et al., 1999).

2.3. Our Model

As is the case for EUD, appropriation implies an independent user who is a domain expert in a specific organizational context. In this specific context, the user can, when carrying out work tasks, exceed the

intentions of an application's designer (McGill, 2004). If the application (e.g. a spreadsheet program) is meant to be used in a flexible way, or is designed to develop applications (e.g. programming tools), it can still be appropriated. This appropriation implies that applications/tools are used to enhance their domain knowledge by developing and continuously redeveloping applications. Applications/tools are appropriated as users discover new opportunities and unanticipated ways to carry out work tasks. The objectives of using the applications/tools can vary substantially but, as a rule, they always cover the features mentioned in the EUD section above: questioning, verbalizing, and decision support.

From the discussion above it is clear that there are close relations between EUD and appropriation. In order to develop a model for the analysis of appropriation in EUD, we have extracted the following proposition, which includes central concepts from the body of knowledge of EUD:

In a specific organizational context, a user(-developer) with a substantial degree of independence and domain expertise can use a tool (e.g. a spreadsheet program) to develop applications that can be used to perform tasks. Such tasks, which can include the verbalization of tacit domain knowledge, and the questioning of existing business activity, serve as decision support and are related to the domain of expertise of the user(-developer) (see Figure 2 below).

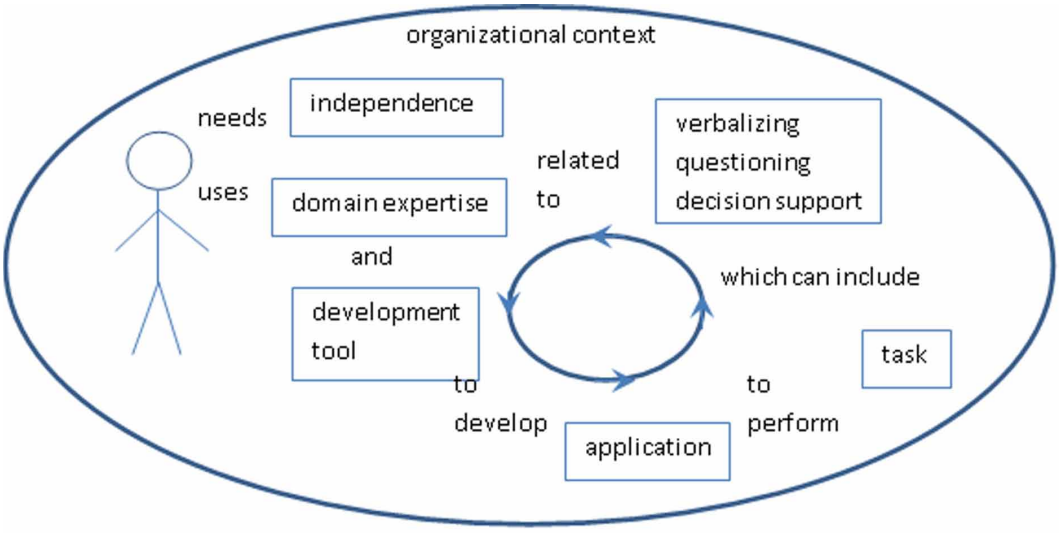
The appropriation dimension is present in terms of the appropriation process (explore, evaluate, adapt) and the design process (analyze, negotiate, design). This is related to EUD via tools and applications. The user-developer uses a tool to develop an application which he/she uses and refines. But the main link is through verbalization, questioning for providing decision support in relation to the user's domain of expertise. This is the driver in EUD as well as in appropriation. The user of a system extends the intentions of the systems designer in order to achieve something in the user's working context. The verbalization, questioning and decision support corresponds to explore-evaluate-adapt and to analyze-negotiate-design. The obvious difference is that appropriation as described by Carroll (2003) the appropriated system is designed by a developer whereas the there are two systems involved in the user-developer appropriation, where one is developed by the user himself.

3. METHOD

In this study, we chose to adopt a case study research method that is explorative, qualitative and interpretative (Walsham, 1995). As shown above, the use of EUD in e-Government has not yet been extensively explored. Indeed, this is the motivation for our use of an explorative approach.

The cases are from Örebro municipality in Sweden. Örebro municipality is the seventh largest in Sweden, with approximately 130,000 inhabitants. Örebro municipality is no different from most municipalities in Sweden; thus, it can be seen as representative of a Swedish mid-sized municipality. Furthermore, it is less necessary for qualitative studies to appear as representative, compared with quantitative studies. So, even though this case was a convenience sample (Oates, 2006), we consider it to be representative. The selection of cases and interviewees was made after asking a systems manager at the municipality's IT department about the use of EUD by civil servants. The manager was deemed to be the most suitable person because he has a major overview of users in the municipality. Our selection criteria were that the users should have practiced EUD for at least five years in order to have gained substantial experience of using EUD and any consequences. Furthermore, they had to have developed applications in a way that made the application of the concept of appropriation meaningful. For example, they had to be actually involved in the development process, not just participate in the evaluation of usability, or contribute with domain knowledge. This implies a relatively independent work situation in which the civil servant is able to decide for himself how to accomplish his work tasks. We were given several suggestions. In addition, we selected four civil servants who fulfilled our criteria. They were from different departments and they all accepted our invitation to participate in the study. After the interviews, we excluded one interviewee because we found out that he developed applications for other people. Thus, he was more of a programmer than a domain expert who develops

Figure 2. Appropriation in EUD



application for himself and his work tasks. Our sampling of interviewees can be classed as purposive because we needed interviewees with solid experience of EUD and appropriation (Oates, 2006).

Data were collected through document studies and interviews. The documents studied were the applications developed by the interviewees: five for A, five for B, and four for C. In some cases, we also studied documentation and reports. The selection of applications for further analysis was based on their potential contribution to the analysis of appropriation. This included whether or not it would be of importance to the civil servant in his work and in some sense typical. With regard to the social welfare planner, we selected applications relevant to social welfare. Likewise, we selected financial applications for the process manager in the finance department, and planning applications for the engineer in the city planning department. While we initially discussed all the applications, later on we focused more on the applications selected for this paper. The final selection of applications presented in this paper was made during the analysis process, because these applications were deemed to be suitable to a discussion of appropriation. The main reason for this was the fact that they were being continuously refined. Furthermore, we preferred to limit the number of applications under discussion in order to study a few applications in more depth. The exception was B, although in his case the selected applications were quite similar and served the same purpose in his work.

Each of the interviewees was interviewed twice, with each interview lasting for 1.5 to 2 hours. The interviews were recorded and transcribed. The main parts of the interviews were conducted by discussing the development process, the developed systems, and their implementation and effects. Questions were asked about development conditions, auditing and documentation. With regard to effects, questions related to functionality, finances, learning, new insights, relations, management, decision support, transparency, unanticipated opportunities and problems. The interviews were semi-structured, with an opportunity for discussions about other topics not previously covered. The transcriptions were then reviewed and approved by the interviewees.

The interviewees provided several applications that were analyzed along with the interviews. Each case is presented in more detail in the results section below. The analysis was qualitative, and was based on six themes. These were derived from a conceptual framework that is made up of both the EUD and appropriation concepts, see motivation of the model in Figure 2 above. The themes were meant to support the interview process and starting with the bigger picture, continuing with specific applications and ending with discussions of effects. The themes are: a) Knowledge Domain,

Expertise And Tasks - background information of the knowledge domain together with the expertise and work tasks of the interviewees, b) Independence And Attitude - knowledge of and attitudes towards the municipality's policies on EUD, and the interviewees' own opinions about the advantages and drawbacks of using EUD in the department, c) Tools And Applications – the software used and the software developed, d) Learning And Decision Support - the interviewees' perceived effects of EUD activities, and e) Appropriation – how users can take advantage of opportunities related to the development and use of applications.

Appropriation as described by Carrol (2003) is mainly concerned with the application development process and does not involve an organizational context. In this study, we are including the organizational context in terms of EUD as well in order to see how the notion of appropriation can support our understanding of EUD. In Table 1 below we depict the relations between the interview themes, EUD and appropriation.

4. RESULTS

In this section we present our case studies.

4.1. Case 1: Planner in Social Welfare

A is a planner in social welfare. A has been working with the development of measurement systems for elderly health care for several years. The social welfare department has 2,800 staff. The vast majority of the staff are enrolled nurses, nurse assistants and secretaries. A handful of administrative staff undertake similar work tasks to A. Indeed, A's work mostly involves the following up of organizational processes. The results of A's work are used as decision support for the elderly health care board. He also provides decision support to departmental managers. Mainly, however, A's work is to follow up on the quality aspects of his department. During the interviews, A was asked to present some applications that were typical to his job. Five examples of applications are described below. We have chosen to describe one of them, "Health care effort", in detail in order to be able to relate to the concept of appropriation in more depth.

Follow up on home care: This application includes the calculated measures and graphical presentation of budget and outcome for 12 geographical areas, distributed between three main categories (i.e., care plan and contact person, complaints, and deviations) and a total of 23 subcategories. Each subcategory has a measure; for example, the number of resource-related complaints or number of medical deviations. The choice of measurements is a crucial matter for A, because it is the basis of the reporting and calculation of "health care effort" discussed below.

Relative use of housing for the elderly: This application is a dynamic table that shows how many beds in elderly housing are used day by day. The numbers are distributed between elderly homes and

Table 1. Relations between EUD, appropriation and interview themes

Themes	EUD (Figure 2)	Appropriation (Figure 1)
Knowledge domain, expertise and tasks	Organizational context, domain expertise, tasks	
Independence and attitude	Independence	Explore, evaluate, adapt
Tools and applications	Application, tools	Technology-as-designed, technology-in-use
Learning and decision support	Verbalizing, questioning, decision support	
Appropriation		Analyze, negotiate, design

different kinds of housing (e.g. for the elderly with dementia, the disabled elderly and so on). These measurements are important for the analysis of resources and also for the calculation of "health care effort".

Quality analysis: In this application, three main categories are calculated and compared from period to period in order to detect any trends that are perceived to represent quality in elderly health care. The categories are: care weight, rejections (of the elderly who apply for care), and care plan + contact person.

Presentation: A stated that the design of the presentation was very important. He had a number of presentations that target different kinds of audiences, such as the health care board, municipality politicians, staff working in elderly health care, colleagues and so on.

Health care effort: In order to find out how to distribute personal resources, A has developed goals and a spreadsheet system for measuring "health care effort"[1], which aims to find a fair measure of the challenges posed by employees' work in the department. Measuring the fulfilment of these kinds of qualitative goals is complicated, because of its nature. Assistance to the elderly can take many forms and include many dimensions. The central system does not cover his information needs. According to A: "The central systems information is not detailed enough. It does not use the same concepts, and the information is not pedagogically comprehensible". The planner's system counts credits for various aspects such as how many caretakers each employee is visiting, how much cleaning work is assigned to each employee, if the employee is visiting caretakers alone or together with a colleague. There are also credits assigned for all kind of medical tasks and circumstances that affects the health care effort. All measures are budgeted and followed up on annual bases. According to A, developing measurements is useful for the learning process of the developer: "When you start measuring a discussion starts". The measurement method is developed iteratively in the form of a spreadsheet application. A does not consider himself to be an expert on the spreadsheet program, but he learns as the need occurs. The spreadsheet program is easy enough for him to use.

In general, the IT department does not encourage EUD activities. However, because existing centralized systems do not produce the information that A needs to develop goals and an effort-measurement system, EUD activities are accepted. There are no plans to develop such a central system because the needs of such a system cannot be specified in a fixed way, making development difficult, at least at present. A: "There is always a need to process data more. It doesn't end". Conditions change, ideas change, the environment changes; thus, systems must also change. A stated that: "It is unthinkable to stop my application development. I need this flexibility, this freedom. The data processing that takes place when you refine the application generates a deeper understanding of the organization. Being a domain expert, this data is not just about numbers. I can translate it into reality".

In his work, A targets municipal politicians and addresses reports to them. The design of reports is a tricky part of his work. It is an informative challenge to find the right level of information granularity.

During his time as a planner, political power in the municipality has changed. In particular, the introduction of new goals has affected the work of the social welfare sector. Although A is developing systems that should be robust enough to manage these new goals, he has turned to national laws and regulations in order to support their robustness. Political change can bring changes to conditions at work. New priorities can generate new goals that need to be operationalized in such a way that they can be followed up, if possible, in a quantitative way. A stated that: "Since this is fundamentally about prioritizing resources, it is a political matter. I must be prepared for changes after an election".

The spreadsheet program was used to develop the first version of a system for the measurement of health care effort. The system was continually developed in a learning process where the learning took place in parallel with the development. As the development and learning process progressed, the system changed in order to correspond to what A learned about the development of the goals and the measurement of health care effort. Finally, these follow-ups were presented to the politicians in an appropriate way. Goals for elderly care cover the overall organization (e.g. average waiting time for a place in care), citizens' opinions (e.g. opinions about participation), finances (e.g. net cost per

patient) and staff opinions about working conditions (e.g. sick leave frequency). According to A, the application process focuses on evidence-based arguments: "There are many opinions and myths based on how it was before. Conditions change. People are getting older. The elderly we serve today were buried some years ago. If we don't base our systems on facts we will not be able to plan properly. Facts change and the central systems are lagging behind." Finally, A stated that it is equally important to have transparent systems: "Transparent systems make the process transparent". A's system can thereby be audited and critiqued, which strengthens the evidence-based character of the goal-setting process.'

4.2. Case 2: Process Manager in Support of Finance Department

B is a financial officer and also the process manager of the finance department, which has 65 employees. He has also worked on feasibility studies in the IT department for several years. Altogether, he has worked on financial matters and IT at the municipality for more than 20 years. Thus, he has substantial in-depth knowledge about his domain.

He confirmed that the municipality does not have an official policy regarding EUD; rather, there is a non-official policy in support of the centralization of all IT. He supports this non-official policy but says that it is impossible not to take on EUD when centralized systems do not cover what is needed. According to B: "End User Development is necessary, and the best use is for prototyping". The obvious drawbacks of EUD are dependence on individuals and poor documentation. An alternative to the total centralization of IT in the municipality would be to use EUD to prototype and specify systems requirements. The main factor in favor of End User Development is "the gap between user needs and existing system support". The central systems are perceived as being rigid and sometimes incompatible with each other. B stated that: "To a certain degree the End User Development processing of complex domain information is also improving the bases for decisions".

B has used a spreadsheet program to create several EUD systems. These systems relate to a variety of investment calculation, cost calculation and cost distribution functions. During the interviews, B demonstrated four applications that he had developed and maintained in order to carry out his work.

Mail management costs: One system is used for calculating how to distribute mail management costs between departments. The main goal here was to identify relevant parameters and give them appropriate weights. Four main categories were established: (1) number of post boxes, (2) number of addresses, (3) post volume and (4) geographical localization. Along with these parameters weights were defined with regard to actual costs. The Swedish post has been privatized and become less regulated; since then, changes to routines, costs, the actors operating the service and so on all have a continuous effect on calculations. Thus, the application must not only contain variable parameters, but also the opportunity to introduce new parameters or delete existing ones.

Planning and calculating ICT investments: Another system has been developed, which is used to plan and calculate ICT investments and their consequences. The goal of this application is to keep track of the financial situation over the next six years and be able to identify the financial space for new investments. It consists of a plethora of parameters that are considered to affect the financial situation. Examples of parameters are: project management, consultancy costs, support, hardware, licenses, operational costs, and so on.

Tenants' rental costs: One system that has been developed by C is a tool for negotiating tenants' rental costs in advance of planned changes. This system presents in detail how rents are calculated. When the rental contract is changed, the costs are specified and agreed upon before signing. "The effect of this is transparency in price setting."

Calculation of systems' costs: Figure 3 below shows calculated costs for systems. These numbers are used to negotiate the distribution of costs between departments. The application is transparent as it is easy to show the formulas behind the totals.

Processing time and storage: The fifth application shows how much processing time and storage the departments have used. The main idea is to provide transparent information as basis for discussions. The department managers can argue for a certain point but the space for guessing and speculating is

Figure 3. Cost calculation example

	A	B	C	D	E	F	G	H
1	OPERATIONAL COSTS PER SYSTEM					ORGANIZATIONAL SYSTEMS		
2 3	Owner	System	Server pool	Computer hall	IT-services	Operation	Storage & backup	Total costs
4	BSU	ADELA	149	81	110	369	37	**746**
5	BSU	Asynja	31	17	0	215	1	**264**
6	BSU	Extens	41	22	6	215	2	**286**
7	Samhb	ALBUM	9	5	0	54	23	**91**
8	Samhb	Klara	9	5	0	54	4	**72**
9	Samhb	BOOKING	36	20	0	54	5	**115**
10	Samhb	BOOK-IT	35	19	95	54	92	**295**
11	Samhb	LMS	18	9	0	54	26	**107**
12	Samhb	IDUS	18	9	0	54	1	**82**
13	Samhb	SILO	127	69	95	215	172	**677**
14	SoV	DocIT	170	92	205	369	130	**967**
15	SoV	PROCAPITA	101	54	95	369	100	**719**
16	Skansli	EMIL	300	163	0	369	530	**1 362**
17	Skansli	KlaPP	10	5	0	54	0	**69**
18	Skansli	PERSONEC	110	60	95	369	210	**844**
19	Skansli	RAINDANCE	169	91	0	369	335	**964**
20	Skansli	Sitevision	85	46	0	369	18	**517**
21	Skansli	Streaming	18	9	0	54	0	**81**
22	Skansli	WDD3	70	38	0	215	7	**330**
23	Vuxam	AMEA	1	0	0	215	0	**216**
24	Vuxam	Progma	1	1	0	215	7	**224**
25	SYSTEMDRIFTKOSTNADER TOTALT		**1 505**	**815**	**701**	**4 305**	**1 701**	**9 027**

heavily reduced when these calculations are accepted as basis for the discussion. When conditions change, it is fairly easy to upgrade parameters and formulas.

The systems mentioned above were developed for informative reasons. According to B: "End User applications can solve the information problem when I present something". Further-more, "People are suspicious about internal transactions. There are lots of opinions". B uses the systems to specify and demonstrate what drives various costs in the municipality; they are particularly useful in discussions and negotiations. The major feature of all B's systems is that they seek to increase the transparency of financial interactions, both internally and externally. A stated that: "When arguments take the form of numbers the discussion becomes more concrete and relevant".

What motivated B to develop these various systems is the inherent flexibility of the spreadsheet program, making it relatively easy to set up a small system in a short space of time. There is a tendency for some of these systems to live on and be developed further as circumstances change in the municipality. EUD has contributed to an increase in the quality of cost measurement. The process has brought the need for dialogue, transparency and evidence-based ways of working. B stated that: "The result is a more realistic view of what are the cost-driving factors".

4.3. Case 3: Engineer at the City Planning Department

C works as a planning engineer in the city's planning department. The department has 130 staff, divided into nine units. C works in one such unit, known as "Sites and exploitation". He has used a spreadsheet program to develop numerous systems of various sizes during his years in the department. As part of his work, he has provided other civil servants and politicians with calculations and presentations within the area of engineering. C stated that: "Some applications are used to support other decision makers and some are for reporting to the management". During the interviews we asked C to demonstrate a typical application from the abundance available to choose from. We will discuss some applications briefly and one system in more depth, because it has interesting dimensions that are specifically relevant when discussing appropriation. C described the following applications:

Investments' effects on rental costs: The application calculates and simulates rents with regard to investment costs. The system is also linked to a number of parameters that represent, for example, interest, indexes and annuity.

Time management: This is a fairly simple application that calculates how much time (and thereby the costs) the various actors spend on different exploration projects. The numbers are distributed in various ways in order to give input that can be used for planning and simulation.

Overview and report: This application compiles several other applications into one overview to give input that can be used for reporting, planning and simulation. It is also possible here to distribute the input in various flexible ways in order to be able to reflect ideas and hypotheses not yet explored. The decision-support systems were built using C's engineering expertise; it would not have been possible to generate those application using central ERPs. C stated that: "Applications can be used for the simulation of various outcomes, which creates flexibility. For example, if we decrease park investments, how does it affect liquidity?"

The site pricing system: Flexibility is also part of the governmental dimension. Following an election in the municipality, the political majority changed. This resulted in several political changes, amongst which was a shift in views on public property. The former social democratic majority held a regulatory viewpoint; however, the new non-socialist majority was more market oriented. This change in political viewpoint resulted in activities that were aimed at selling off some municipal property.

This political change, which occurred at the time of the study, meant that the work of the engineer largely came to include responsibility for developing pricing rules for the sale of public properties and sites. This task was part of public administration; thus, any sales had to be carried out for the social, economic and environmental benefit of the citizens. As a public servant, C was committed to maintaining the goals of rational public administration, which were achieved by making the politicians responsible, agree the rules for selling with a relevant and complete decision-support system. To achieve all these goals, decision making had to take into account the follow-up and assessment of liquidity, and the evaluation of properties and business ratios for assessment of profit. In his role as engineer, C used a spreadsheet program as the development tool. The system (see Figure 4 below) was used for assigning the price of sites owned by the municipality of Örebro. The price was decided by assigning values to a number of variables. "We had no idea from the beginning how to decide a valid price." The main factors were site location and building potential. With regard to site location, this included: orientation, attraction, nature, and geography. The main building prerequisites were shape, topography, and terrain. In Figure 4 the factors are to the left. Each factor is assigned a value. In the figure example, the factor orientation is assigned one point. Various calculations add up the points. To the left a table shows the value of sum of points, e.g. four points equals 1,50. Finally the formulas of the application add up to a price per square meter, which is shown in the bottom right corner. The arrangement of the factors and their calculations were arranged so that factors could be reviewed one by one, thereby providing opportunities for structured discussions. According to C: "My domain expertise, in combination with a flexible tool, made it possible to finally agree upon a price setting model. In fact, the system formalizes existing knowledge regarding the domain of site pricing factors".

The calculated costs were also adjusted according to the road contract index to ensure they matched the actual costs of the municipality. The development of the spreadsheet system enabled the results of the political change to be implemented, because the politicians had to define precisely the consequences of the method used for valuing the sites. Various versions of the system were presented to the politicians, helping to stimulate discussions in the political arena. The model was interactive (as described above); thus, simulations could be made at the meetings, e.g. by changing the factors or points that affected the final price. The engineer participated at the meetings and helped the politicians to make changes to the system during the discussions. When the discussions drew to a close, a pricing policy was agreed upon by the city council. The system formed part of the basis of their decision.

Figure 4. Site pricing system

	A	B	C	D	E	F	G	H	I	J	K
1	**Support for property evaluation**										
3	**PLACE**	Orientation, attractivity (acc to plan), nature values, central/peripheral									
4		PLACE					SUM				
5	CLASS	Orientation	Attractive	Nature	Geography		Total			Points	Factor
6	1-3	1	2	1	2		6			4	1,50
7										5	1,40
8	1(0) = best				FACTOR:		1,3			6	1,30
9	3 = worst									7	1,20
10	Site close to commuting street:			+ 1						8	1,10
11	Entry street south side:			2						9	1,00
12	New exploitation in a field:			3						10	0,90
13	Old garden etc:			1						11	0,80
14										12	0,70
16	**Building potential**	shape of site, topography, grounds									
17											
18		POTENTIAL TO BUILD					SUM			Points	Factor
19	CLASS	Shape	Topography	Foundation			Total			3	1,00
20	1-3(-7-ground	1	3	5			9			4	0,95
21										5	0,90
22	1 = best				FACTOR:		0,7			6	0,85
23	3 (7) = worst									7	0,80
24	Use of filling material (foundation			<= 3 p						8	0,75
25	Pile-driving (foundation):			4-7 p						9	0,70
26	Short site:			<= 2 p						10	0,70
27	Long site:			<= 5 p						11	0,70
29	Time of evaluation		07-07				Basic price SEK			**100 000**	
30	INDEX [Date]		1,14				Area Price SEK/sqm			**103 740**	

During the lifetime of the system, occasional changes were made to the model. Various adjustments had to be made as sites were sold. All together the application was flexible and transparent. The interaction and real-time adjustments (during meetings with the politicians) were possible due to the flexibility and openness of the spreadsheet program.

The system can be related to several of the municipality's goals. Firstly, the municipality should be non-profit making, and buyers should be dealt with fairly and openly. Secondly, although prices could be adjusted to match the actual costs of the municipality, a price should be assigned that represents the market value, so that citizens are dealt with fairly.

Whilst C can see that his applications have drawbacks, such as the complexity of huge spreadsheets, he does not blame the spreadsheet application for any risks and failures. C stated that: "It is not the systems that produce information. It is me. I'm always responsible for evaluating the correctness of the outcome".

5. ANALYSIS

The goal of our appropriation model is to find out how appropriation takes place in EUD. We used the model presented in the Theory section, which can be formulated as follows: In a specific organizational context a user(-developer) with a substantial degree of independence and domain expertise can use a tool, such as a spreadsheet program, to develop applications. These can be used to perform various

tasks, including the verbalization of tacit domain knowledge, and the questioning of existing business activities. They also act as decision support, and are related to the domain of expertise of the user(-developer) (see Figure 2 above).

Our analyses of the individual variables of the model is given below.

5.1. Organizational Context

The organizational context of this study is a Swedish municipality in which there are various departments administered by civil servants. Being a civil servant means that you are accountable to the citizens of the municipality. It also means that you have to follow various laws and rules specific to the public sector; these relate to democracy and participation, as well as privacy and integrity. In the public sector of a democratic country, its servants have to adjust to the contemporary political majority. If the ruling party/parties are replaced after an election, goals and policies might change. This dimension is only indirectly related to appropriation; instead, it is more directly related to EUD in the public sector.

5.2. Independence and Domain Expertise

A necessary condition for end user appropriation is professional independence combined with domain expertise. All three interviewees were all skilled professionals within their domains. They also all enjoyed independence in their work, meaning that they were free to choose how they accomplished goals related to their work. Obviously, stricter rules would have meant there was no room for experimenting and innovating solutions to various problems. Stricter rules would also hamper the appropriation process, as we define it.

5.3. Development tool

Spreadsheet programs were used in all our cases. Compared with e.g. standard package, a spreadsheet program is highly appropriable because it was meant to be used for development. In our specific context and with the domain expertise of the user-developer, access to certain data, and awareness of the preferences of the receivers of information, it forms a specific development context in which appropriation is cultivated.

The appropriability of a tool is dependent on how difficult it is to master that tool. To develop some of the spreadsheet applications there was no need to be highly skilled or experienced. On the other hand, one could say that the more the user-developer knew about the tool, the more he could achieve with regard to the goals of his domain. In terms of affordance, the spreadsheet tool provided capabilities to the user-developers that were perceived as affordances.

In general, the potential for goal achievement appears to be correlated with the development skill of the user-developer. However, from a tool skill point of view, the health care effort application developed by A is very straightforward. The main asset of this application is the way in which it operationalizes the goals of effectiveness, efficiency, fairness and transparency. It is notable that the application and its underlying logic were distributed to all parties involved. It is also worth noting that the simulation potential in these systems was frequently used in order to refine the applications as much as possible. Indeed, simulation can be seen as a formalization of the negotiations that took place in all of the cases. It is even possible for the receivers of the information (i.e. the politicians) to manipulate the underlying algorithms to fine tune the application according to their own preferences. This is due to the appropriable nature of the spreadsheet program.

5.3. Application

The applications developed by the user-developers in the study vary substantially in terms of complexity and purpose. They do, however, have one aspect in common: they were used to fulfill the goals of the user-developer himself.

In this study, A provided decision support to politicians in the form of elaborated goals for healthcare efforts. Among other things, B developed tools for negotiating costs with various departments in the municipality. Finally, C developed tools that have an engineering dimension, and which are used for decision support at various levels within the municipality. The goals of all three civil servants were to measure efficiency and effectiveness in elderly care, internal finances, and engineering-related tasks. In actuality, they all operationalized policies and goals that are relevant to their expertise and experiences. They used the applications as tools to accomplish these goals.

Furthermore, the applications were not considered as complete or final at any stage. Rather, they were in a state of constant development, reflecting the status of contemporary knowledge of the domain in question. The applications were actually a dialectic part of the dialog/negotiation between different parties in different domains. At some point, the applications were used as the basis for decisions. But this didn't mean that, later, they were not further elaborated upon. In general, there was a mix of development and maintenance; thus, the applications could be seen to represent the current state of knowledge in a specific department. Both the tool (first order) and the applications (second order) are appropriated in a way that makes them transparent and useful for the accomplishment of goals.

5.4. Task

In the public sector, tasks are expected to be handled with accountability and, unless they involve aspects of privacy, transparency. Municipalities are diverse in nature; all kinds of tasks, from firefighting and transport to health care and family counseling, are carried out in them. All of these tasks must be handled according to laws and regulations and, at the same time, be transparent and accountable. However, in many respects, times are volatile. Contemporary conditions tend to change, leading to new opportunities and risks, all of which have to be taken into consideration. All of the civil servants presented in this study are exposed to these volatile contemporary times. In addition to the political context, which can change every fourth year, IT systems are replaced and updated, departments and units are reorganized, new techniques, contracts and agreements are implemented, and new financial restrictions or models are agreed upon. Together, they constitute the user-developers' everyday situation. So, when the user-developer decides to develop an application to meet his needs he is already aware of the public context and the tasks that exist within it; thus, he can take this context into consideration when developing the application. The need to carry out tasks using ICT tools, such as spreadsheet programs, is the basis for appropriation. The description above is specifically related to EUD in the public sector.

5.5. Verbalizing, Questioning and Decision Support

As discussed above, the applications developed by user-developers more or less represented the status of the dialogue between the parties involved. At the same time, in all the cases, the applications were used as tools to question contemporary knowledge status. If everything was known in the domains there would be no need to investigate these projects and routines. The main motive for user-developers is actually a desire to extend knowledge about heath care effort through the measuring of cost distribution, and site pricing. Furthermore, the drive to know more is combined with the need to provide better decision support for the betterment of the management of the municipality. Together, they help to enforce the overall goals of the municipality. It is important here to understand that the applications are not just ICT artifacts; rather, they represent the actual tasks, goals and processes of the user developer's knowledge domain. The application is a medium through which existing tacit and explicit knowledge can be verbalized and questioned in order to provide decision support. If the tool and the application are appropriable, they will support this process.

5.6. Appropriation

Appropriation is defined as taking advantage of opportunities related to the development and use of applications, when the developer has in-depth knowledge of the problem domain and is also the

primary user of the application. Below, we present an extended and revised version of the appropriation model put forward by Carroll (2003) (see Figure 1 above).

We differentiate between tools and applications, where tools are used to develop applications. In the case of spreadsheet applications, the tool (i.e., a spreadsheet program such as MS Excel) and its functions are available in the application both in development and in use. If we refer to case C and the site pricing system, the developer could use spreadsheet functions and formulas both during and after the development of this application. The tool offers ongoing opportunities to use available resources to add and revise functions when necessary. Thus, the user-developer can take advantage of opportunities and affordances because of his in-depth domain knowledge and an appropriable tool. Both tool (e.g. the spreadsheet program) and application (e.g. the site pricing system) have their own different but still related cycles of learning. The more the user-developer learns about the tool the more options and affordances are at hand when developing and appropriating the application. Thus, there is a parallel appropriation of the tool and the application.

Appropriation begins with evaluating and learning the tool. This phase is affected by at least three factors. One such factor is (1) the interaction the user-developer might have in his community of user-developers, which includes co-workers or colleagues in other organizations who work in a similar way. As for user-developers, there are several discussion forums and similar interaction hubs where it is possible to get useful information on how to achieve certain goals. Another factor that affects a user-developer is (2) information from the tool provider or new versions of the development tool. New versions bring new opportunities that can then be appropriated. For example, when Microsoft included Pivot tables in Microsoft Excel, a new tool for analysis and reporting became available. Finally (3) the IT department and its support staff can act as assistants or provide useful knowledge on any IT-related matter. Together, these three factors form a hotbed for ideas and solutions to problems that user-developers may encounter. Such problems may affect the appropriability of both the tool and the application.

First order appropriation takes place when developers use the development tool to develop an application and continue to use and evaluate the tool. Second order appropriation takes place when the user-developed applications are used, evaluated and maintained. When end users develop, use and maintain systems using a spreadsheet program they are appropriating the affordances of two systems simultaneously. The first is the spreadsheet program, whilst the second is the application. The user must control a minimum of the tool's capabilities in order to be able to develop a more or less complex application. When using or maintaining (appropriating the affordances of) the application, the spreadsheet program is still present as an integral part of the application. Its functions (formatting, diagramming, pivot tables, etc) are available when using or maintaining the application, no matter how simple or complex, thus providing affordances to the user. The first order appropriation, i.e. making the spreadsheet program a natural tool for the user, is necessary for the second order appropriation, i.e. being able to refine and take continuous advantage of the affordances of the application. Continuously the user evaluates both the spreadsheet program and the application with regards to organizational context and goals. In this process, the user can appropriate more affordances of the spreadsheet program and also appropriate the application in terms of continuous maintenance developing second order affordances.

From our point of view, a spreadsheet program can constrain or afford an end user (Leonardi, 2011). In the cases presented in this paper, end users perceive the affordances of their spreadsheet programs and applications because of human agency (Giddens, 1984). The end users are goal driven; they perceive such affordances because they support their efforts to develop and achieve their goals. According to Leonardi, people perceive technology as having different constraints or affordances, with different effects (Leonardi, 2011). The fact that spreadsheet programs are highly appropriable because of their unrestricted nature supports the affordance alternative. As for our cases, the user-developers sought new ways of acting, new routines and even new goals, with user-developed applications as the means. An important factor is the in-depth knowledge of the user-developers, which belongs to

human agency. They were constantly aware of the specific constraints of a public civil servant. They were also independent enough to be able to form and formalize goals within the public norms. This is revealed both in terms of the "care effort" goals and the site pricing application.

Notable here is that the EUD appropriation context is decentralized, individual, and continuously refined. Appropriation in general deals with centralized systems, multi-user environments, and dedicated tools. The main difference between the two is the decentralized end user context, which provides possibilities of affordances that are generated directly from the end user's goals and preferences and not via a system developer or an IT-department. In spite of all risks related with EUD, this direct connection between the user and his/her tool provides a possibility to deal with professional, tacit, unarticulated knowledge of the user, Appropriation is an appropriate label for this specific process, where the tacit dimension can be present throughout the development process and not being forced to be formulated (and reduced) by the need for formalization when describing requirements to a IT-specialist.

Appropriation becomes more significant when we consider evaluation activities. All user-developers evaluated their applications together with clients, employers, politicians, co-workers or someone with the ability to assess the value of the application. The user-developers took advantage of the tool's features as well as those of the application. They used fairly simple functions, formulas and commands. Combined with their deep domain expertise and independent work situations, they were able to form the basis for questioning and decision support.

6. DISCUSSION

In this section, we discuss our findings with regard to appropriation. All our cases include developers who have a major in-depth knowledge of the organizational context in which they work. The problem domains differ, but are all closely related to local government. The crucial concept here is opportunity. Does EUD support our developers so that they can take advantage of, or even detect, opportunities?

In a way, it seems as if appropriability, in terms of potential to be used for application development in various contexts, is the major feature of a tool such as a spreadsheet program. The more the user learns about the program, the easier it becomes to develop applications to support task activities. This is surely the case for A in the first case. The applications developed are in themselves drivers, not just for the user-developer, but also for the involved parties. The operationalization of health care quality affects the entire department, not to mention the patients and their relatives. The rest of the cases fulfil the same role within their task context.

The reason appropriation of end user tools takes place at all is that the user-developers simply have to carry out tasks needed but not supported by central systems. It takes time in terms of maintenance of central systems to include functions for what is formulated as an information requirement by a user. When and if it is done, new requirements appear. Thus, it is a flow of new requirements where the end user development alternative is instantly at hand to support a public or private servant in need of a certain report or a specific calculation. The reason for avoiding ERP systems may not be that such systems lack functionality. It may be that middle managers still prefer to develop "feral systems" because of their mistrust of centralized ERP systems (Kerr & Houghton, 2008)

Appropriation takes place when applications and their development are integrated in the work context and its tasks. Actually, the tasks are carried out via the applications and the applications are developed in order to carry out tasks. The tools offer the opportunities that assist the user-developer to achieve his goals by enhancing his action repertoire. In the first case, A was able to elaborate on his measurement of health care effort and other qualitative aspects of elderly health care. In the second case, B provided a transparent basis for the negotiations of costs. In the third case, C provided a decision tool for politicians so that they could agree upon a transparent model for site pricing. These tools could also be manipulated by clients, employers or other involved actors during discussions. The findings are clearly in line with those of Fahy and Murphy (1999), who argued that "the process of

systems development helped the managers verbalize and amplify the understanding of the situations they faced" (Fahy & Murphy, 1999, p. 197). The combination of domain expertise and tool skills enables the user-developer to question and enhance his understanding of his own knowledge domain (Avdic, 2009). In a centralized environment, this way of working is not possible.

To summarize, why does appropriation take place? The basic reason is the lack of central system support. Central systems can't cover all of a diversified municipality's needs. Central systems are not perceived as effective tools and do not have enough capabilities to provide the kind of affordances needed by civil servants, who need to question, negotiate, explore, improve and inform politicians, managers, colleagues and citizens. As for the three cases in our study, a spreadsheet program was perceived as just such a tool. The spreadsheet system is appropriable (first order) because of its nature, which allows the user-developer to use his up-to-date in-depth domain knowledge, together with a flexible tool, to develop applications that support existing business goals. The developed applications are in turn appropriable (second order) when they function as explorers in the ever-changing volatile reality of a diversified municipality today. Some applications are very simple, although size is not important here. The mail cost distribution system developed by B is an example of an application whose main role is to form the basis for the distribution of costs. When compared with ordering a development job from the ICT department or an external consultant (if the latter is allowed at all), the effort involved in developing the application and at the same time identifying the requirements is considerable. The application itself becomes an integral part of B's work context. At the same time, it is part of a number of similar applications used to carry out B's job. The development process of an application can generate an idea that can be implemented in a similar system. This is second order appropriation; the applications are appropriable.

Obviously, there are risks associated with allowing users to develop their own systems. Risks concern mainly relate to data integrity, security and maintenance (Panko & Aurigemma, 2010; Powell et al., 2009). These risks must be taken seriously and be balanced against the opportunities offered by EUD. Stories have been told about companies that have lost millions of dollars as a result of erroneous spreadsheets (Wailgum, 2007); nonetheless, we consider this risk to be one of many that are related to the work of experts in the private and public sectors. Regardless of the medium, people make mistakes and perform low quality calculations. Education and auditing can prevent not just erroneous EUD applications but also many other problems that also occur in more traditional systems development. Further on

So far, there has been little research activity into public sector EUD in the form described in this paper. We believe this has to do with the informal shape of EUD. Even though it covers an important aspect of IT processing, namely what cannot be covered by the existing, centralized systems, it is still not perceived as official. Official or not, EUD will not go away. We need to take advantage of its benefits and eliminate its drawbacks, because a flexible world will always need appropriable tools. This leads us to consider some policy implications for EUD in the public sector and the practices of end user-developers. Education about the risks and their avoidance is an obvious way to go. The implementation of support and auditing techniques are also necessary measures (Mehandjiev et al., 2006). Policies can also be stricter and firmly regulate the use of end users' tools. Finally, in order to reduce risks, there are some interesting developments on tools for end users. Kuttal et al. (2014) promoted the possibility of a versioning add-in for end users to organize their maintenance activities. One recurring idea put forward is to support end users with frameworks, models and methods in order to organize EUD and minimize errors. Cunha et al. (2014) proposed, "a method for extraction of a relational model from a spreadsheet and the subsequent embedding of the model back into the spreadsheet to create a model-based spreadsheet programming environment". Hermans et al. (2012) tested an approach that uses "code smells" (i.e., symptoms that indicate problems) for error detection, translating the findings to data flow diagrams. Jannach et al. (2014) classified existing "automated spreadsheet quality assurance (QA) approaches, which range from spreadsheet visualization, static analysis and quality reports, through to testing and support and model-based spreadsheet development". They also proposed an

outline for future automated spreadsheet quality assurance. Compliance management is yet another strand of systems development organization with regard to risk management. It comprises the design, implementation, maintenance, verification and reporting of compliance requirements that originate in the regulations and in law enforcement. (Ramezani et al., 2012). According to Abdullah et al. (2012), compliance management "…is referred to as the coordinated set of activities designed to assure that all elements of the business (processes, employees, partners, and assets) strictly follow any established regulatory requirements" (Abdullah et al., 2012, p. 429). There are also ISO standards that may be relevant for EUD risk management, i.e., ISO 31000:2009 (implementation of risk management) and ISO 27005:2012 (guidelines for information security risk management). Another potentially relevant framework for EUD risk management is the Enterprise Risk Assessment version of COSO (Committee of Sponsoring Organizations of the Treadway Commission). The framework can be used to evaluate the risks to which an organization is exposed.

Obviously, there are ways of anticipating or remedying the risks associated with EUD. How could these methods and frameworks have been applied in the cases described in this paper? Attempts to reduce such risks by organizing EUD as described above can be seen as an example of transferring the methodological traditions of traditional systems development to the EUD domain. Would it be realistic to implement automated spreadsheet assurance, compliance management, ISO risk management or even COSO in Örebro municipality and target all, or at least most developers in order to reduce risks? To some extent these organized efforts tend to constrain the flexible and innovative dimension of EUD, whereby the user-developer is free to develop applications at any time for any relevant reason. Whatever the method or framework used, it is worth noting the words of C, who said that it is he, as a professional engineer and civil servant, who is responsible for the consequences of his applications, not the spreadsheet program. Thus, future research could focus on calculating and comparing the consequences of various risk-reducing approaches.

7. CONCLUSION

This paper has sought to respond to the research question: How does appropriation take place in the public sector in the development of end user applications by civil servants?

We conclude that public servants with deep problem domain knowledge can take advantage of end user tools, such as spreadsheet programs, in the problem-solving process in order to meet vaguely defined problems. Appropriation is manifested in the continuous development of various applications; for example, tools for quality control, cost distribution, and the reporting of site price information. The functions of the developed applications were not available via the central systems.

In parallel with the application development process, a learning process also takes place. During the process, end users gain more in-depth knowledge and a greater understanding of their own organization by questioning and analyzing facts and assumptions. Appropriation is enabled by these parallel processes. The performance of EUD in a public-sector context adds an analytical dimension that is not easily achieved in a centralized environment. First order appropriation takes place when the user-developer uses and evaluates the development tool, which in our case is a spreadsheet program, making it a personal tool for addressing problems in his domain. If the tool has capabilities that are perceived as affordances by the user-developer, the tool is appropriable. The use of the tool is actually the development of applications that target problems in the domain of the user-developer. When these applications are maintained because of changes in the surrounding world or the ideas of the user developer, the applications are appropriated. This is termed second order appropriation.

Limitations of the study are the number of user-developers. We have prioritized the detailed description of cases over the quantity of cases. However, we have access to numerous examples that are just as good examples of end user appropriation in both the public and the private sector.

Future research could build on the appropriation model generated in this study. In so doing, they could refine the model and include an awareness of the negative aspects of EUD. Overall, we

identified a lack of research about appropriation in EUD in the public sector. Thus, researchers can contribute with cases, techniques, methods and theories that support the creation of tools for knowledge development by EUD.

REFERENCES

Abdullah, N. S., Sadiq, S., & Indulska, M. (2012, January). A compliance management ontology: Developing shared understanding through models. In *Advanced Information Systems Engineering* (pp. 429–444). Springer Berlin Heidelberg. doi:10.1007/978-3-642-31095-9_28

Alter, S. (2014). Theory of Workarounds. *Communications of the Association for Information Systems*, *34*(55), 1041–1066.

Avdic, A. (2009). Spreadsheet End User Development and Knowledge Management. In M. Koshrow-Pour (Ed.), *Encyclopaedia of Information Science and Technology* (2nd ed., pp. 3564–3572). Hershey, NY: Information Science Reference. doi:10.4018/978-1-60566-026-4.ch568

Berthelsen, O. W., & Zander, P. (2006). Appropriation in the Development of Information Systems For Voluntary Organisations. In *Proceedings of the 14th European Conference on Information Systems (ECIS 2006)*, Goteborg, Sweden.

Buonanno, G., Faverio, P., Pigni, F., Ravarini, A., Sciuto, D., & Tagliavini, M. (2005). Factors affecting ERP system adoption: A comparative analysis between SMEs and large companies. *Journal of Enterprise Information Management*, *18*(4), 384–426. doi:10.1108/17410390510609572

Carlo, J. L., Lyytinen, K., & Boland, R. J. (2012). Dialectics of Collective Minding: Contradictory Appropriations of Information Technology in a High-Risk Project. *Management Information Systems Quarterly*, *36*(4), 1081–1108.

Carroll, J. (2004). Completing design in use: closing the appropriation cycle. In *Proceedings of the 12th European Conference on Information Systems (ECIS 2004)*, Turku, Finland.

Carroll, J., Howard, S., Peck, J., & Murphy, J. (2003). From adoption to use: The process of appropriating a mobile phone. *AJIS. Australian Journal of Information Systems*, *10*(2), 38–38.

Cunha, J., Mendes, J., Saraiva, J., & Visser, J. (2014). Model-based programming environments for spreadsheets. *Science of Computer Programming*, *96*, 254–275. doi:10.1016/j.scico.2014.02.002

Dawes, S. S. (2009). Governance in the Digital Age: A research and action Framework for an uncertain future. *Government Information Quarterly*, *26*(2), 257–264. doi:10.1016/j.giq.2008.12.003

Errors, S. (2010). Retrieved 2015-12-08 from http://panko.shidler.hawaii.edu/SSR/

EuSpRiG. (2015a). European Spreadsheet Risk Interest Group. Retrieved 2015-12-08 from http://www.eusprig.org/

EuSpRiG. (2015b). EuSpRiG Horror stories. Retrieved 2015-12-08 from http://www.eusprig.org/horror-stories.htm

Fahy, M., & Murphy, C. (1999). Managers, Information and Systems Development: Exploring the Tacit Dimension. In *Information Systems at the Core: European Perspectives on Deploying and Managing Information Systems in Business* (pp. 188–204). Dublin: Blackhall.

Faraj, S., & Azad, B. (2012). The Materiality of Technology: An Affordance Perspective. In P.M. Leonardi, B.Nardi & A. Kallinikos (Eds.), Materiality and Organizing: Social Interaction in a Technological World. Oxford University Press Oxford.

Firman, S. (2015). Microsoft Excel Versus Apple's Numbers: Who prevails? *Excelwithbusiness*. Retrieved 2015-12-08 from https://excelwithbusiness.com/blog/post/microsoft-excel/microsoft-excel-versus-apple-s-numbers-who-prevails-

Fogli, D., Colosio, S., & Sacco, M. (2010). Managing accessibility in local e-government websites through end-user development: A case study. *Universal Access in the Information Society*, *9*(1), 35–50. doi:10.1007/s10209-009-0158-z

Gaskin, J., & Lyytinen, K. (2010). Psychological Ownership and the Individual Appropriation of Technology. In *Proceedings of the 43rd Hawaii International Conference on System Sciences*. doi:10.1109/HICSS.2010.472

Gebauer, J., & Ginsburg, M. (2009). Exploring the black box of task technology fit. *Communications of the ACM*, *52*(1), 130–135. doi:10.1145/1435417.1435447

Gibson, J. J. (1977). The Theory of Affordances. In R. Shaw & J. Bransford (Eds.), *Perceiving, Acting, and Knowing: Toward an Ecological Psychology* (pp. 67–82). Hillsdale, NJ: Lawrence Erlbaum Associates.

Giddens, A. (1984). *The constitution of society: Outline of the theory of structuration*. Univ of California Press.

Goel, S., Dwivedi, R., & Sherry, A. M. (2014). Process Alignment, End User Participation for E-Government Programs: Key Stakeholders View. *International Journal of Public Administration in the Digital Age*, *1*(2), 65–79. doi:10.4018/ijpada.2014040104

Grönlund, Å. (2000). *Managing Electronic Services: A Public Sector Perspective*. London: Springer-Verlag. doi:10.1007/978-1-4471-0511-4

Hermans, F., Pinzger, M., & van Deursen, A. (2012). Detecting and visualizing inter-worksheet smells in spreadsheets. In *Proceedings of the 34th International Conference on Software Engineering* (pp. 441-451). Piscataway, NJ: IEEE. doi:10.1109/ICSE.2012.6227171

Houghton, L., & Kerr, D. V. (2006). A study into the creation of feral information systems as a response to an ERP implementation within the supply chain of a large government-owned corporation. *International Journal of Internet and Enterprise Management*, *4*(2), 135–147. doi:10.1504/IJIEM.2006.010239

ISO. (2009). ISO 2015 ISO 31000:2009 Risk management -- Principles and guidelines. Retrieved 2015-12-08 from http://www.iso.org/iso/catalogue_detail?csnumber=43170

ISO/IEC. (2011). ISO/IEC 2700, 2015 ISO/IEC 27005:2011 Information technology — Security techniques — Information security risk management (2nd ed.). Retrieved 2015-12-08 from http://www.iso27001security.com/html/27005.html

Jannach, D., Schmitz, T., Hofer, B., & Wotawa, F. (2014). Avoiding, finding and fixing spreadsheet errors – A survey of automated approaches for spreadsheet QA. *Journal of Systems and Software*, *94*, 129–150. doi:10.1016/j.jss.2014.03.058

Jones, M. C. & Twidale, M. B. (2005). What's in a name? Exploring the connections between abstraction and appropriation. *International reports on socio-informatics, 2*(2), 43-47.

Karhanna, E., Strub, D. W., & Chervany, N. L. (1999). Information Technology Adoption Across Time: A Cross-Sectional Comparison of Pre-Adoption and Post-Adoption Beliefs. *Management Information Systems Quarterly*, *23*(2), 183–213. doi:10.2307/249751

Karlsson, F., Holgersson, J., Söderström, E., & Hedström, K. (2012). Exploring user participation approaches in public e-service development. *Government Information Quarterly*, *29*(2), 158–168. doi:10.1016/j.giq.2011.07.009

Kerr, D. (2014). Feral Information Systems and Workarounds: The Present Position. Feral Information Systems Development: *Managerial Implications*, *23*.

Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., & Wiedenbeck, S. et al. (2011). The state of art in end-user software engineering. *ACM Computing Surveys*, *43*(3), 21. doi:10.1145/1922649.1922658

Kuttal, S. K., Sarma, A., & Rothermel, G. (2014). On the benefits of providing versioning support for end users: An empirical study. *ACM Transactions on Computer-Human Interaction*, *21*(2), 9. doi:10.1145/2560016

Leonardi, P. M. (2011). When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. *Management Information Systems Quarterly*, *35*(1), 147–167.

Leonardi, P. M. (2013). When does technology use enable network change in organizations? A comparative study of feature use and shared affordances. *Management Information Systems Quarterly*, *37*(3), 749–775.

Lepouras, G., Vassilakis, C., Halatsis, C., & Georgiadis, P. (2007). Domain Expert User Development: Smartgov approach. *Communications of the ACM*, *50*(9), 78–83. doi:10.1145/1284621.1284623

Lieberman, H., Paterno, F., Klann, M., & Wulf, V. (2006). End-User Development: An emerging Paradigm. *Human-Computer Interaction Series*, *9*, 1–8. doi:10.1007/1-4020-5386-X_1

McGill, T. (2004). The Effect of End User Development on End User Success. *Journal of Organizational and End User Computing*, *16*(1), 41–58. doi:10.4018/joeuc.2004010103

Mehandjiev, N., Sutcliffe, A., & Lee, D. (2006). Organizational view of end-user development. In *End user development* (pp. 371–399). Springer Netherlands. doi:10.1007/1-4020-5386-X_17

Meyer, S., Vogel, J., & Mehandjiev, N. (2012). An end-user friendly service delivery platform for the public sector. In *Case Studies in Service Innovation* (pp. 141–145). Springer New York. doi:10.1007/978-1-4614-1972-3_28

Microsoft. (2013). Microsoft Excel Help Guide.

Oates, B. J. (2005). Researching information systems and computing. *Sage (Atlanta, Ga.)*.

Oliveira, T., & Martins, M. F. (2011). Literature Review of Information Technology Adoption Models at Firm Level. *The Electronic Journal Information Systems Evaluation*, *14*(1), 110–121.

Panko, R. R. (1998). What we know about spreadsheet errors. *Journal of End User Computing*, *10*(2), 15–21. doi:10.4018/joeuc.1998040102

Panko, R. R., & Aurigemma, S. (2010). Revising the Panko-Halverson taxonomy of spreadsheet errors. *Decision Support Systems*, *49*(2), 235–244. doi:10.1016/j.dss.2010.02.009

Pankowska, M. (2010). Development of the strategy of Information Communication Technology. In J. Korczak, H. Dudycz, & M. Dyczkowski (Eds.), *Advanced Information Technologies for Management -AITM 2010* (pp. 218–232). Wroclaw: Publishing House of Wroclaw University of Economics.

Perry, J., & Rainey, H. (1988). The Public-Private Distinction in Organization Theory: A Critique and Research Strategy. *Academy of Management Review*, *13*(2), 182–201.

Pipek, V. (2005). *From Tailoring to Appropriation Support: Negotiating Groupware Usage* [PhD Thesis]. University of Oulu, Finland.

Polanyi, M. (1967). *The Tacit Dimension*. New York: Anchor books.

Powell, S. G., Baker, K. R., & Lawson, B. (2009). Impact of errors in operational spreadsheets. *Decision Support Systems*, *47*(2), 126–132. doi:10.1016/j.dss.2009.02.002

Ramezani, E., Fahland, D., van der Werf, J. M., & Mattheis, P. (2012, January). Separating compliance management and business process management. In *Business Process Management Workshops* (pp. 459–464). Springer Berlin Heidelberg. doi:10.1007/978-3-642-28115-0_43

Säljö, R. (2000). *Lärande I praktiken. Ett sociokulturellt perspektiv*. Stockholm: Prisma.

Scaffidi, C., Shaw, M., & Myers, B. 2005. Estimating the Numbers of End Users and End User Programmers. In *Proceedings of the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)* (pp. 207–214).

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User Acceptance of Information Technology: Toward a Unified View. *Management Information Systems Quarterly*, *27*(3), 425–478.

Wailgum, T. (2007). Eight of the Worst Spreadsheet Blunders. *CIO*. Retrieved 2015-12-08 from http://www.cio.com/article/2438188/enterprise-software/eight-of-the-worst-spreadsheet-blunders.html

Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems*, *4*(2), 74–81. doi:10.1057/ejis.1995.9

World Bank. (2011) *eGovernment*. Retrieved 2015-12-08 from http://web.worldbank.org/WBSITE/EXTERNAL/TOPICS/EXTINFORMATIONANDCOMMUNICATIONANDTECHNOLOGIES/EXTEGOVERNMENT/0,contentMDK:20507153~menuPK:702592~pagePK:148956~piPK:216618~theSitePK:702586,00.html

Wulf, V., & Jarke, M. (2004). The Economics of End-User Development. *Communications of the ACM*, *47*(9), 41–42. doi:10.1145/1015864.1015886

ENDNOTES

[1] In Swedish, "vårdtyngd"

Anders Avdic is Assistant Professor in Information Systems at Dalarna University, Sweden. He received his PhD in Information Systems in 1999 from Linköping University. His research focuses IT and learning in general and specifically End User Development, Knowledge Management and IT in Pedagogy.