Tailoring Tools for System Development

Anders I. Mørch
University of Bergen, Norway

This paper gives a summary of end-user tailoring, which is a method for system development that starts not from low-level building blocks nor high-level frameworks but from intermediate-level generic applications. By starting from something that already works, end-user tailoring echoes a concern heard in many organizations today: infrastructure for information technology is in place, but the information technology itself is not well adapted to the tasks the organizations wish to use it for. This concern is addressed in this paper from a technical as well as a user-oriented perspective: integrating tailoring tools into generic applications and evaluating the usability of the tailoring tools with end-users.

End-user tailoring is a method for system development that starts from existing systems and further develops them to adapt them to needs that were not accounted for in the original systems. The existing systems addressed in this work are referred to as generic applications. Examples of generic applications are word processors, e-mail systems, and drawing programs. Previous studies has shown that the majority of users of a commercial word processor (92%) do tailor their applications in various ways, such as setting parameters in preference forms to customize the system (Page et al., 1996). However, this is not always easy for end-users; and to make it easier the authors recommend that tailoring tools be better integrated with generic applications.

A claim made in this paper is that to make tailoring easier, the transition from using a system to tailoring it should be made seamless, and to accomplish this I propose that tailoring tools be better integrated with generic applications. The interplay between tools and tasks plays an important role in this paper, and is in part motivated by how tools evolve to meet new needs. Petroski (1992) has illustrated this by examples of how everyday artifacts, such as knives and forks, evolve to fit their tasks. Knives and forks, he shows, have gradually developed to simplify work (cutting and eating) and to adapt to new needs as well (eating behavior). Since most users are familiar with everyday artifacts, I use a can opener as an example throughout the paper (Figure 1). It serves as an illustrative analogy grounding the concept of generic application in everyday terms. The analogy is further developed later in the paper.

The paper is organized as follows. First, I describe the background for end-user tailoring. Then, I recommend a level of abstraction at which it should be addressed. Next, I propose a suite of tailoring tools that is based on the level of abstraction. Such a suite of tools has been built by the author and integrated which an application is being used. Several authors have pointed out the inevitable interdependence of tasks and tools (e.g., Norman, 1988; Carroll & Rosson, 1992), which includes the observation that when tools are improved, the nature of the tasks tends to change as well, often making the tasks simpler (Norman, 1988).

The paper gives a summary of end-user tailoring, which is a method for system development that starts not from low-level building blocks nor high-level frameworks but from intermediate-level generic applications. By starting from something that already works, end-user tailoring echoes a concern heard in many organizations today: infrastructure for information technology is in place, but the information technology itself is not well adapted to the tasks the organizations wish to use it for. This concern is addressed in this paper from a technical as well as a user-oriented perspective: integrating tailoring tools into generic applications and evaluating the usability of the tailoring tools with end-users.

End-user tailoring is a method for system development that starts from existing systems and further develops them to adapt them to needs that were not accounted for in the original systems. The existing systems addressed in this work are referred to as generic applications. Examples of generic applications are word processors, e-mail systems, and drawing programs. Previous studies has shown that the majority of users of a commercial word processor (92%) do tailor their applications in various ways, such as setting parameters in preference forms to customize the system (Page et al., 1996). However, this is not always easy for end-users; and to make it easier the authors recommend that tailoring tools be better integrated with generic applications.

A claim made in this paper is that to make tailoring easier, the transition from using a system to tailoring it should be made seamless, and to accomplish this I propose that tailoring tools be better integrated with generic applications. The interplay between tools and tasks plays an important role in this paper, and is in part motivated by how tools evolve to meet new needs. Petroski (1992) has illustrated this by examples of how everyday artifacts, such as knives and forks, evolve to fit their tasks. Knives and forks, he shows, have gradually developed to simplify work (cutting and eating) and to adapt to new needs as well (eating behavior). Since most users are familiar with everyday artifacts, I use a can opener as an example throughout the paper (Figure 1). It serves as an illustrative analogy grounding the concept of generic application in everyday terms. The analogy is further developed later in the paper.

The paper is organized as follows. First, I describe the background for end-user tailoring. Then, I recommend a level of abstraction at which it should be addressed. Next, I propose a suite of tailoring tools that is based on the level of abstraction. Such a suite of tools has been built by the author and integrated which an application is being used. Several authors have pointed out the inevitable interdependence of tasks and tools (e.g., Norman, 1988; Carroll & Rosson, 1992), which includes the observation that when tools are improved, the nature of the tasks tends to change as well, often making the tasks simpler (Norman, 1988).

The interplay between tools and tasks plays an important role in this paper, and is in part motivated by how tools evolve to meet new needs. Petroski (1992) has illustrated this by examples of how everyday artifacts, such as knives and forks, evolve to fit their tasks. Knives and forks, he shows, have gradually developed to simplify work (cutting and eating) and to adapt to new needs as well (eating behavior). Since most users are familiar with everyday artifacts, I use a can opener as an example throughout the paper (Figure 1). It serves as an illustrative analogy grounding the concept of generic application in everyday terms. The analogy is further developed later in the paper.

The paper is organized as follows. First, I describe the background for end-user tailoring. Then, I recommend a level of abstraction at which it should be addressed. Next, I propose a suite of tailoring tools that is based on the level of abstraction. Such a suite of tools has been built by the author and integrated which an application is being used. Several authors have pointed out the inevitable interdependence of tasks and tools (e.g., Norman, 1988; Carroll & Rosson, 1992), which includes the observation that when tools are improved, the nature of the tasks tends to change as well, often making the tasks simpler (Norman, 1988).

The interplay between tools and tasks plays an important role in this paper, and is in part motivated by how tools evolve to meet new needs. Petroski (1992) has illustrated this by examples of how everyday artifacts, such as knives and forks, evolve to fit their tasks. Knives and forks, he shows, have gradually developed to simplify work (cutting and eating) and to adapt to new needs as well (eating behavior). Since most users are familiar with everyday artifacts, I use a can opener as an example throughout the paper (Figure 1). It serves as an illustrative analogy grounding the concept of generic application in everyday terms. The analogy is further developed later in the paper.

The paper is organized as follows. First, I describe the background for end-user tailoring. Then, I recommend a level of abstraction at which it should be addressed. Next, I propose a suite of tailoring tools that is based on the level of abstraction. Such a suite of tools has been built by the author and integrated which an application is being used. Several authors have pointed out the inevitable interdependence of tasks and tools (e.g., Norman, 1988; Carroll & Rosson, 1992), which includes the observation that when tools are improved, the nature of the tasks tends to change as well, often making the tasks simpler (Norman, 1988).

The interplay between tools and tasks plays an important role in this paper, and is in part motivated by how tools evolve to meet new needs. Petroski (1992) has illustrated this by examples of how everyday artifacts, such as knives and forks, evolve to fit their tasks. Knives and forks, he shows, have gradually developed to simplify work (cutting and eating) and to adapt to new needs as well (eating behavior). Since most users are familiar with everyday artifacts, I use a can opener as an example throughout the paper (Figure 1). It serves as an illustrative analogy grounding the concept of generic application in everyday terms. The analogy is further developed later in the paper.

The paper is organized as follows. First, I describe the background for end-user tailoring. Then, I recommend a level of abstraction at which it should be addressed. Next, I propose a suite of tailoring tools that is based on the level of abstraction. Such a suite of tools has been built by the author and integrated which an application is being used. Several authors have pointed out the inevitable interdependence of tasks and tools (e.g., Norman, 1988; Carroll & Rosson, 1992), which includes the observation that when tools are improved, the nature of the tasks tends to change as well, often making the tasks simpler (Norman, 1988).
into the generic application BasicDraw. BasicDraw has been tailored by the author to develop specialized applications. An example of this is given. Finally, the usability of the tailoring tools has been studied in an informal user experiment. Results from the experiment are reported.

Background

Although household appliances, such as the can opener, are relatively easy to use, they are not easy to modify. Hardware appliances are difficult to modify because the building blocks are static and not meant to be modified (black boxes). Software applications, on the other hand, have a greater potential for modification because they are built out of dynamic building blocks (programs). However, software modifiability does not come for free because reprogramming the building blocks can be difficult. Care must be taken to find the proper balance between program/machine efficiency and user/task proficiency. End-user tailoring tries to achieve this by combining programmable applications with end-user participation. This is further motivated next.

Programmable Applications

Programmable applications (Eisenberg, 1995) and design environments (Fischer & Girgensohn, 1991; Eisenberg & Fischer, 1994) are classes of computational environments that give an end-user support in modifying and extending the environments. Modifiability is supported by the access to high-level, domain-oriented building blocks (analogous to LEGO bricks), and extensibility is accomplished by writing small programs in dynamic programming languages (e.g., LISP and JAVA).

A goal of programmable applications is to support programming not in terms of low-level machine instructions but in terms of high-level building blocks and to elevate programming to an activity that approaches the use of the applications (Fischer and Girgensohn, 1991). A comprehensive discussion of some of the motivational issues behind programmable applications is given by Mørch (1997a).

In field studies of end-user programming behavior, Nardi (1993) has observed that users of programmable applications are able to master the formal notations embedded in these systems (scripting languages of spreadsheets and CAD systems), when the notations match the tasks the users wish to perform. This is in part because users are already familiar with formal notations from other activities they engage in, such as knitting and baseball scorekeeping.

End User Participation

Involving end-users as first-class participants together with developers and managers in projects where new technology is introduced is a well-established principle in Scandinavia (Nygaard, 1986). Users are the foremost experts on their own work and therefore need to be involved in decisions regarding the introduction of new technology at their workplace, since the technology will often change their existing ways of working.

Ehn and Kyng (1984) suggest a “tool perspective” for how to address development and use of computer systems. This perspective emphasizes that end-users be active participants in the development process rather than passive suppliers of data to “fill in” design specifications. The authors recommend that users participate in the making of design specifications by means of tools and materials they are familiar with from other activities - at home or at work (Ehn & Kyng, 1991).

Bjerknes and Bratteteig (1984) propose an “application perspective” in order to involve end-users in design. The application perspective sees technology from the context of use, i.e., the users’ work tasks. These tasks are different from the formal procedures encoded in the technology they use to accomplish the tasks. On this basis, the authors recommend that computer systems be built by studying the use context and of finding new and creative ways to use existing technology to challenge and influence the development of technology.

Scandinavian end-user participation has primarily been concerned with involving end-users in the early stages of systems development, i.e., during analysis, design and implementation. The perspective presented this paper is that end-user participation should not stop once a system has been installed, but continue to involve end-users in future developments as well (Braa, 1995; Mørch, 1997a). This is addressed in the present paper by combining the tool perspective and the application perspective: end-user tailoring is both a method for integrating tailoring tools into generic applications and an approach for involving end-users in systems development.

End User Tailoring: Concepts and Tools

User Interface Objects

The interface of a computer application is composed of user interface objects. User interface objects mediate interaction between a user and a system in order to get work tasks done. The work includes tasks such as writing reports,