

# From Implementation to Design: Tailoring and the Emergence of Systematization in CSCW

*Randall H. Trigg*  
Xerox Palo Alto Research Center  
3333 Coyote Hill Road  
Palo Alto, CA 94304  
USA  
Tel: +1-415-812-4863  
E-mail: trigg@parc.xerox.com

*Susanne Bødker*  
Department of Computer Science  
Aarhus University  
Ny Munkegade 116  
DK-8000 Aarhus C  
Denmark  
Tel: +45-89423256  
E-mail: bodker@daimi.aau.dk

## ABSTRACT

In this paper, we look at how people working in a governmental labor inspection agency tailor their shared PC environment. Starting with standard off-the-shelf software, the tailors adapt that software to the particular workplace in which they are embedded, at the same time that they modify and extend the practices of that workplace. Over time, their adaptations and the tailoring processes themselves become structured and systematized within the organization. This tendency toward systematization is in part a response to the requirement that the results of tailoring be sharable across groups of users. Our study focuses on several dimensions of the work of tailoring: construction, organizational change, learning, and politics. We draw two kinds of lessons for system development: how better to support the work of tailors, and how system developers can learn from and cooperate with tailors.

**KEYWORDS:** Tailoring, customization, emergent use of standard technology, development and use of shared standards.

## INTRODUCTION

The world of computer system development and use is increasingly characterized by blurred boundaries. No longer are "independent," "isolatable," and "monolithic" the code-words for success. Today's systems must survive in an environment populated with other technologies and systems, the selection and behavior of which can rarely be controlled or predicted in advance. Similarly, the strict boundaries be-

tween technologies and situations of use are being relaxed. Today, there is growing recognition that a "system" includes the institutional, organizational, and workgroup context around an installed technology. Finally, the traditional binary (and techno-centric) division of people into designers and users is blurring. Increasingly, we recognize a rich spectrum of people having a variety of skills from the application of computer technology in everyday work to the design of new technology.

Furthermore, we are investing increasing human and technological resources at and on those boundaries. For example, product organizations compete in the marketplace based in part on the flexibility of their systems. This can involve the number of platforms their software runs on, the degree to which the product can be "integrated" with other software, and the customizability of their systems by users having little or no programming skills. Witness, for example, the number of systems delivered with Application Programming Interfaces (APIs).

In communities of everyday users of technology the blurred designer/end-user distinction is the basis for a new community of practice [16], people who behave like designers as well as users, and in whose hands the success and long-term survival of an installed technology often rests. Researchers studying the phenomenon use labels like tinkerer, translator, and gardener [10, 17, 18]. These studies have taught us much about the work of these user-designers and its vital importance, and have led to recommendations for system designers and managers wanting better to enable such "tailoring."

In this paper, we present a study of such an emergent community of practice. In particular, we have studied (and participated in) the work lives and "everyday" dilemmas of four people dealing with new technology in an organization not primarily concerned with system building. Being embedded

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.  
CSCW 94- 10/94 Chapel Hill, NC, USA  
© 1994 ACM 0-89791-689-1/94/0010..\$3.50

in the organization, they share professions and practices with their fellow employees. In addition, they confront problems raised by a set of technologies recently and currently being installed. Of course, many people working in organizations encounter computer systems in the course of their work. What sets these four apart, however, is that their attempts to address technology problems and needs are (to varying degrees) *on behalf of* the organization or group in which they work.

Wendy Mackay called these tailors "translators" and proposed that managers give them official recognition and status [17]. Thereafter, Michelle Gantt and Bonnie Nardi [10] found that certain CAD organizations already had such positions. They used the CAD designers' own terms, "gurus" and "gardeners," to distinguish local experts from tailors having official institutional recognition.<sup>1</sup> Our work can be seen as continuing the enterprise started by Mackay, Nardi, and others. What distinguishes this study from some of this work, however, is that we are concerned not only with the technical tailoring activities of local developers and the attendant activities of support, maintenance, and training. We also want to understand the work life of local developers as it is defined and constrained by the organizational and institutional contexts in which they are embedded. Thus, for example, we have found the classification by MacLean et al. [18] with its primary focus on technical skills to be far too limited.<sup>2</sup>

The tailors in our study work in an organization where shared procedures and consistent outcome is of the utmost importance. This is at the core of what makes tailoring in such a real-life organization "computer supported cooperative work," rather than an anarchistic exploration of technological possibilities by individuals or groups. Thus tailoring, usually seen as enabling an ever-increasing variety of use patterns, can play an equally important role in organization-wide efforts to standardize, bounding the variety that the technology in principle makes possible.

Contrast this with Brown and Duguid's [8] notion that tailorability leads to "social demassification." They seem to argue that tailorability inherently works against people's ability to share objects and conventions. As we show in this paper, quite the contrary is the case in this workplace. It is precisely the tailorability of their PC software and the efforts of their local developers that allow the organization

to institute and reify the standard practices that support Brown and Duguid's "shared borders." We see the activities of sharing and distribution not as counter-measures to fend off creeping demassification, but rather as natural parts of the organizational process of adapting and appropriating technology.

Following brief descriptions of our project, the setting we have been working in, and four tailors from that setting, the heart of the paper discusses the work of tailoring from the points of view of construction, change, learning, and politics.<sup>3</sup> We conclude by suggesting that the activities of tailoring and of "real" system development can profitably inform one another. Our understandings of the development work of tailors can help us design systems and tailoring environments that better fit their needs. Furthermore, we believe that understanding better the deeply embedded nature of tailors' work can inform and positively influence the practice of system development.

### The Empirical Study

This study is part of a larger research project on system development conducted cooperatively with a local branch of the Danish national labor inspection service, hereafter referred to as the AT.<sup>4</sup> Underlying the project is the principle that computer system design should be about helping users of technology change their work situation for the better. System development should be organized as a learning process where the participants, collectively and as individuals, improve their ability to understand and manage processes of technological and organizational change (e.g., [11]). Our interest in coupling technical and organizational issues also means that the research questions we address derive in part from particular problems confronted at the AT. Our participatory research and design strategy is discussed further in [4, 6].

Two years before the time of this writing, the AT introduced PCs running WordPerfect and Windows. We followed the use of this technology from its early introduction, conducting informal interviews and observations of everyday work in the Århus branch office, and focusing on a small group of early users. For six months one of the authors (Bødker) spent approximately a half day every third week at the branch. The days were picked so as to fall at varying times during the week. It was agreed that she would help with technical problems as they arose, in exchange for time spent talking to the inspectors and secretaries.<sup>5</sup>

<sup>1</sup>Okamura et al [23] refer to the officially sanctioned local developers they studied in a Japanese R&D lab as *mediators*.

<sup>2</sup>MacLean et al [18] classify technology users into workers, tinkers, and programmers. According to this classification, the workplace we discuss here consists of workers, a few tinkers, and one programmer. Our four tailors belong in the latter two categories, although they certainly also qualify as workers.

<sup>3</sup>This framework follows Andersen et al's [1] analysis of the work of system developers.

<sup>4</sup>AT is short for "Arbejdstilsynet", translated literally, "The Work Inspection."

<sup>5</sup>MacLean et al [18] characterize this as the "handyman's" role.

Later, after six months of less frequent interaction with the AT, we conducted interviews with three labor inspectors and the programmer now in charge of software and hardware. The hour-long interviews focussed specifically on tailoring and sharing of tailored programs. The interviewer (Bødker) was by this time well acquainted with all four interviewees. Though the quotes and certain other material appearing here stem from the interviews, the paper as a whole is based on the understandings of the AT that we gained over the full four-year course of the project.

### The Setting

The AT is a state institution that inspects and advises companies on health and safety matters. Until the mid-1970s, it conducted mostly physical work environment inspections in factories. The Work Environment Act of 1975 brought a more holistic view of the work environment and broadened the scope to include non-factory work. The act also changed the professional profile of the inspector; therapists and psychologists were hired, and prevention became a central issue. In the late 80s came further decentralization, a client orientation, and quality assurance through upward accountability in the bureaucracy. (For a discussion of the changing role of the AT in Danish society, see [19].)

The use of PCs, WordPerfect, and Windows started in 1992 with a small group at the Århus branch consisting of eight inspectors and a secretary. One year later, the use of the technology spread to all of the approximately 50 inspectors. Accompanying the technological change was a crucial change in work practice: secretaries stopped writing for the inspectors. Today, inspectors produce their own texts and perform most of their own information retrieval work. In addition to word processing software, they have access to e-mail and some central data bases from their PCs, but run almost no other computer applications.

Tailoring at the AT primarily involves customizing WordPerfect with button panels, macros, standard forms, and paradigm (collections of legally valid standard phrases). The integration of WordPerfect with other applications is, however, a matter of serious concern. The integration effort has concentrated up to this point on central databases, such as the AT's nationwide "VIRK" database, running on a VAX and containing information about companies and inspections. (For an analysis, see [5].) Underlying these efforts is the infrastructural work of making network, printers, and workstations function together, carried out by AT staff and external consultants.

### The People

Overall, some 50 inspectors and 10 secretaries work at the Århus branch of the AT. Most of these have varying amounts of experience with WordPerfect/Windows ranging from none at all to over a year of doing their own word pro-

cessing. They share an interest in the computer, but only to the degree that it helps them in their work. Only a few have modified the version of WordPerfect/Windows provided for them.

The technical staff consists of a temporary programmer and two of the inspectors (referred to here as DA and DB) who have official local developer status.<sup>6</sup> In addition to their work as labor/health inspectors, DA and DB manage and modify computer technology used at the AT. Our interviews with these two local developer inspectors revealed that DA is mainly responsible for PC technology while DB covers the VAX/mainframe applications. We also interviewed a third inspector (referred to as T), an advanced user of the technology which he also sometimes tailors. However, he does not have official local developer status nor is his tailoring work primarily for the benefit of his colleagues. Finally, we interviewed the branch's programmer (referred to as P) who does some tailoring in addition to his other system support and training duties.

Each of our four protagonists work on the borders between technology development and everyday work at AT, although to different degrees and along different dimensions. P, working full-time on system support, is closest to the technology development side, and is less rooted in the community of practice of labour inspection. At the same time, his (for him, somewhat isolated) position at the AT places him in the midst of an organization with no interest in technology development beyond that required to support the work of inspectors, lawyers, and administrators. At the other extreme among these four is the inspector T. Though not an official local developer, he has a relatively advanced understanding of the technology and is able to craft his technological environment to some degree. He adds buttons to his WordPerfect panels, but doesn't write macros, believing they are beyond his reach as well as that of most users in the organization. Inspectors DA and DB, the officially-recognized local developers, lie somewhere in between. They are given time and resources to pursue the technological well-being of the organization. DA and DB, beyond their formal responsibility for maintaining the equipment, have a personal interest in the technology, and spend time at work and at home exploring it. They have customized their versions of WordPerfect/Windows and have designed button panels, macros, and standard forms for their colleagues to use.

Table 1 outlines the responsibilities covered by our four protagonists. While (initially) unfamiliar with everyday

---

<sup>6</sup>We use the term 'local developer' to indicate that their tailoring work is not only approved by management, but is actually part of their job description. According to Bonnie Nardi's [21] terminology DA and DB would be called 'gardeners.'

work at the AT, P has the most experience with the technology. Of the four, he is the only one engaged in "programming."<sup>7</sup>

	P	DA	DB	T
Official AT inspector	no	yes	yes	yes
Official AT tailor	yes	yes	no	
Tailors for the benefit of others	yes	yes	yes	no

Table 1. Tailoring responsibilities

### The Work of Tailoring

Initially DA and DB had full responsibility for maintaining and tailoring the equipment, but a year after obtaining the PC's and network, P was hired and given overall responsibility for computer support at AT-Arhus. At the time of our investigation, he had worked in the organization for a year. Currently, most of the tailoring is channeled through him; anyone with a problem and a proposed solution can ask him to help implement it. In practice, the ideas usually come through DA or DB; after attempting to solve a problem on their own, they pass their partial solution along to P who addresses any remaining technical problems. The result then comes back to DA or DB who try it out and often modify it further. As DA put it when asked about his tailoring responsibilities in the wake of P's hiring:

Yes, formally [the tasks have moved to P], but what makes [a solution] really clever, or at least the last detail, is hard to come by.

DA had experienced how a solution made by P wouldn't run on most people's computers because they didn't have access to the appropriate directories. The example illustrates the local developers' sensitivity to how the modifications will work in practice, from the overall level of knowing what role a document plays in labor inspection, down to finding that a macro puts a document in a folder that is inaccessible to ordinary users. Tailoring is only a part of the larger effort of *adapting* technology by and for particular users and their work practices [26]. Our local developers' tailoring efforts should be seen in the context of their overarching concern with work at the branch. Indeed, exploring, tailoring, integrating, and otherwise adapting the technology to the work being conducted all around them has made DA and DB, the organization's official mediators and articulators between design and use.

<sup>7</sup>See Nardi [21] for a discussion of the concept of "programming" in the context of technology tailoring.

Table 2 indicates the range of tailoring skills. DA's tailoring skills are especially apparent in his use of WordPerfect. Consider for example, the branch's "paradigm" collection: a systematic application of WordPerfect button panels to support the reuse of standard textual phrases. DA, DB, and T all manipulate the buttons making up these panels. P, DA, and DB also create such buttons. DA, in addition, has written button-controlling macros. Though DB tailors his own PC environment to some degree, his work as a local developer has focused on the branch's mainframes where he has set up standard ways of extracting information from databases. Finally, DA and DB struggle to integrate WordPerfect with other applications, an area where they feel that they quickly reach the limits of their competence.

	P	DA	DB	T
Installs WordPerfect buttons	yes	yes	yes	yes
Builds/modifies WordPerfect buttons	yes	yes	yes	no?
Writes WordPerfect macros	yes	yes	no?	no
Tailors mainframe software	no?	no	yes	no
Integration	yes	yes	no	no
Programs on the PCs	yes	no	no	no

Table 2. Tailoring skills

In summary, DA and DB, in close cooperation with P, form the heart of a new, emergent, community of practice of tailoring at the AT. DA and DB are, and will likely remain, peripheral to a wider community of systems developers and technicians who develop and adapt WP applications, and to the technology design community in general. (We return to the question of the technical isolation of the AT tailors in the conclusion.)

### TAILORING AS CONSTRUCTION

Though the work of tailoring is multi-faceted, it has construction at its heart. The following example is typical of the construction work of tailoring at the AT and illustrates the ways that systematization gradually becomes a part of the process.

As the Arhus branch of the AT grew accustomed to having PCs on the desks of the inspectors, the work process changed in various ways. For example, the forms that had been filled out by secretaries, using mainframe text processing and a special printer, were now to be handled by in-

spectors from their PCs. Early in the process of adapting WordPerfect to these new work practices, DA designed a standard online form which, when printed, would fit on AT's official pre-printed "claims" form. His initial brute force realization used manually inserted spaces and linefeeds to lay out the page. The fragility of this scheme quickly became apparent as other inspectors tried to use DA's form; it was far too easy to inadvertently destroy the layout, say, by deleting a linefeed or by inserting extra space. During one of our visits we watched T, one of the more advanced users of WordPerfect, run up and down the hall, printing and reprinting, before obtaining a result that satisfactorily fit on the pre-printed form. As a result, DA began to search for a more robust solution. After experimenting unsuccessfully with various WordPerfect features, he finally turned to two computer science students taking part in our investigation. They helped him arrive at a solution based on WordPerfect's *schema* concept.<sup>8</sup>

As is typical in such cases, DA started with a particular problem, needing to print online documents onto the claims form. But a degree of robustness was required in order for his solution to work for his colleagues as well as himself. A schema-based solution offered the possibility of such robustness, but required significant knowledge from the designer/tailor and might never have happened without assistance. Building WordPerfect schemas requires inserting and manipulating otherwise invisible "codes" in the document, a familiar notion for those used to working with, say, Unix-based document formatters, but unfamiliar to those used to working with WYSIWYG text editors. (The learning required of local developers in such cases is discussed in a later section.) What started as ad hoc but goal-directed tailoring grew more systematic in response to the requirement that the artifact be sharable.

Another case of technology tailoring emerged from the evolving practice of mailing out standard letters. Previously, the branch had sent out letters on at least five different pre-printed forms. Though a few of their printers had multiple paper trays, none had more than three. In order to avoid changing the contents of paper trays for each print job, they chose a single standard paper form with pre-

printed letterhead and logo, and let the computer generate the remaining format.<sup>9</sup>

Though this solution worked for many of the forms, one in particular caused problems. Here, the back of one of the copies of the sheet had been used as a reply or acknowledgment form providing space for the recipient's signature. This made replies from "customers" identifiable to the AT when the forms were returned. Because the standard form they now generated was blank on the back, AT lost the standard "look and feel" of the replies as well as the specific information. For example, companies now might forget to put case numbers on replies. DA chose to tailor WordPerfect so that the creation of these forms would trigger the creation of an additional reply form. After some amount of work (including learning about WordPerfect macros), he managed to write a macro that not only automatically generated a reply page, but partially filled it in with information (e.g., identifying the particular case) extracted from the letter the reply page was to accompany.

Here again, a problem arose as others tried to use DA's macro. The macro built the reply page at the same time the document was first created. However, the inspectors frequently started by copying an existing document and changing the case identification. In such cases, the macro did not run, and the reply page was printed with the old case identifier. After some trial and error, T (a user of DA's new form) got around this problem by modifying his method of document re-use. Rather than start with a copy of an existing document, he created a new document for the new case (causing the proper reply page to be generated), and then pasted in text from the existing document.

It is easy to imagine fixes for DA's "bug," say, a macro that recomputes the contents of the reply page on command rather than only on document creation. But the lesson here is deeper: the ability (and the luxury) to step back and anticipate the need to run the macro at different times is not normally part of the tailor's repertoire. On the contrary, the style of system development taught and encouraged among professionals, moving from analysis to design to realization (e.g., [1]), does not and probably cannot apply to the tailor's world.<sup>10</sup> For practical as well as competence reasons, tailors start with concrete realizations, using generalization from experience to work backwards toward (re)design and analysis.

<sup>8</sup>Though an improvement over the manual scheme, this solution had problems as well, for example, with text wrap-around. In addition, even after building the new form and learning the new concepts, DA was left with a kind of "nested" tailorability problem: the users of DA's new form could not themselves modify the schema as they could with the old brute-force version. DA was required to be personally involved in even minor cases of form modification.

<sup>9</sup>They considered printing the AT letterhead and logo from the computer, however, this required purchasing color printers which they couldn't afford.

<sup>10</sup>In truth, many system developers no longer follow a strict analysis-design-implementation model. Nonetheless, an emphasis on the importance of up-front reflection, modelling,

## MANAGING CHANGE

The processes of change prompted by system development/tailoring at the AT take three forms: technical changes to the software and hardware of computer systems; changes to work practices accompanying the introduction and modification of technologies; and changes to the activities of managing technology change. In what follows we focus on the third of these.

As Wendy Mackay [17] has argued, the work of tailoring is more than the modification of technologies in service of user needs. A crucial part of the process is sharing and distributing these changes. At the AT, significant effort has been put into finding appropriate ways of distributing what are called "standards" among the workers. This term is used at the Aarhus branch to refer to changes and additions to the software that are meant to be used throughout the branch. These include paradigms, new or modified buttons and button panels, new WordPerfect forms and schema for generating them, and new or modified macros.

The sharing/distribution of standards happens in three ways:

1. Paradigms are proposed and debated in meetings of a technology committee. After approval by the branch's lawyers, they are distributed (primarily by P) throughout the branch.
2. WordPerfect schemas and forms are designed by individual workers as well as by local developers. Usually, only the schemas end up being distributed to others.
3. Macros and button panels are developed by P, DA, and DB, and distributed throughout the branch.

At first, standards (forms, macros, etc.) developed and spread opportunistically. Someone heard about tailoring done by a colleague (often DA or DB), copied their modifications, and perhaps performed further customizations of their own. Now the process is more systematic; ideas are conveyed to the programmer and local developers, who use them as the basis for new standards. Distributing standards at the AT is semi-automatic; individual PC's are configured to download the new facilities when booted each morning.<sup>11</sup> When potentially disruptive changes are downloaded, the workers are explicitly notified and told how the new functionality is intended to be used. Except for certain modifications related to the technical infrastructure (e.g., the network), people at the

---

and conceptualization remains fundamental to both the rhetoric and the practice.

<sup>11</sup>This strategy works well for the time being, although we foresee problems with updates that conflict with people's own modifications, a phenomenon well-known from more advanced PC installations.

AT are free to use or ignore the standards they receive.<sup>12</sup> In any case, they normally do not make standards of their own independent of this process.

The tailors we talked to were pleased with the new process, arguing that it gives equal access to standards throughout the branch, improves the quality of the standards, and eases their own work by ensuring consistency across the branch. In the case of paradigms (standard branch-wide phrases and legalese), a structured review process is also in place: the technology committee meets regularly to discuss the suggestions collected by P and to decide which to adopt. (The politics around this committee will be discussed later.)

The increased structuring and bureaucratization of the tailoring process can also be seen in the now formally-recognized roles of the tailors and in the identification and assignment of certain technology-related responsibilities to the "programmer" job. Thus, the need-driven process of systematization described earlier is echoed at the organizational level in the movement from tailoring as an ad hoc activity to its current status as a vital part of the organizational infrastructure.

Tailoring, usually seen as enabling an ever-increasing variety of use patterns, can play an equally important role in organization-wide efforts to standardize, bounding the variety that the technology in principle makes possible. Our use of the term "standardize" underscores the role tailoring has in reifying in software the procedures and forms that characterize the work and the output of the AT. We see standardization and situated/negotiated problem solving as mutually constitutive (see e.g., [12, 24]). Standards are crystallizations of procedures that emerged from situated problem solving [3]. In reifying such standards, the work of tailoring contributes to their being locked into organizational practice. In addition, the systematization of the processes of tailoring further constrains the organization of work while it enables (and requires) ever more advanced tailoring.

## LEARNING

For tailors and others at the AT, learning happens in conjunction with technical and organizational change. On the one hand, over time, tailors learn the technical aspects of macros and other elementary programming constructs they call "tricks."<sup>13</sup> At the same time, tailors sometimes find themselves responsible for the learning of their co-workers. (This is reflected by the title "computer instructor" bestowed by AT on the local developers DA and DB.) In

---

<sup>12</sup>The freedom to ignore the change has the disadvantage of reducing the kind of feedback tailors need from their users.

<sup>13</sup>See Nardi & Miller [20] for a discussion of the need-driven manner in which tailors learn programming concepts.

what follows, we first consider the role of tailors in the technology-related learning of their co-workers, and then the tailors' own learning process and the ways in which it is fundamentally collaborative.<sup>14</sup>

### Learning How to Use the Technology

Technology-related learning at the AT occurs both informally and formally. Informal learning is usually triggered by a problem encountered by someone at the branch when working with the technology. Sometimes they ask for help from their physically nearest co-workers. Often, however, they request help from particular inspectors. The tailors described for us the network of who-asks-whom. Those at the branch with the most technical experience seek advice directly from the programmer or the local developers. Less knowledgeable workers have a "guru" (not one of our tailors) to whom they turn when help is needed.<sup>15</sup> As T explained:

I ask P or DA. Probably those of us who are most advanced ask P. Others ask us, it is like a staircase. R asks me, but I don't ask him, there are different stairs.

Formal learning happens in training courses conducted by teachers from outside the AT. Periodically, people are asked what their needs are, and courses are offered to cover those needs. DB and P are responsible for selecting people for external training. According to DB,

P and I have taken on computer training in a structured way. We are uncovering what it is people here in the branch ought to be able to do, and which people should be able to do it--at what level. Take macros.... Some people should be capable of developing them, though most will only use them.... If nobody here can develop them, then we have to pay to have them developed.

Thus, learning at the AT occurs through a combination of structured, but informal learning networks and formal training. Again, the AT tailors are at the heart of both activities.

### Learning to Tailor

Learning for the tailors is largely by trial and error. As Miller and Nardi [20] describe, it is opportunistic, driven by their own needs and those of co-workers in the organization. Furthermore, the learning of our AT tailors has a distinctly collaborative character. Indeed, echoing

<sup>14</sup>The term "mutual learning" is sometimes used to emphasize the dual learning required by both designers and users in system development (e.g. [14]). We see an analogous phenomenon occurring in the tailoring context.

<sup>15</sup>This network is similar to the patterns of unix customization file distribution described by Wendy Mackay [17].

Engestrom [9], the tailoring expertise itself can be said to be interactive.

Moreover, advanced tailoring is itself a cooperative process. Consider three of our tailors: P, DA, and T. P cannot tailor completely on his own, because of his limited understanding of AT work practices. DA's concerns grow out of the need for his tailoring work to support others in the organization. Where T mainly uses already-tailored technology as a platform, DA works with, and through WordPerfect, calling on P for help as needed. Tailoring expertise at the AT resides as much in the long-term cooperative efforts of DA and P, and T and DA, as it does in any of them individually. The way they support and inform each other's learning--T and DA toward greater levels of technical competence and P toward a greater appreciation of AT work practice--is a crucial part of adapting technology at the AT.

### POLITICS AND ETHICS

Tailoring at the AT can be seen as a political process in which goals are negotiated based on differing and sometimes conflicting perspectives, power and resources. The political aspects of the process can be hidden, or out in the open and under discussion. In what follows, we discuss the political circumstances of the tailors' work both within the Arhus branch and across the AT as a whole.

### The "Technology Committee"

Politics plays a role in how tailoring happens and how its products move through the local organization. For example, as we noted earlier, decisions to adopt and distribute paradigms for the branch are made in a "Technology Committee." This committee is a place where diverse interests meet. Management wants to increase productivity and at the same time insure that the wording of the AT's directives is above legal reproach. The lawyers want to control the legalese the inspectors use, preferring that only "professional" phrases be fixed in the computer. And while the inspectors agree on the importance of writing directives that will hold up in court, they want the freedom to craft language that addresses the particular problems raised in each case. The committee provides a forum for dealing with clashes among these interests.

Note that it was the tailorability of WordPerfect that motivated and justified the creation of this forum. The fact that buttons could be customized to insert particular bits of text into WordPerfect documents led managers and lawyers at the AT to confront the question of *which* texts. The tailorability of the technology first led to the development of systematic means of distributing changes, which in turn led to the creation of formal committees to manage those changes. The benefit (at least from management's point of view) is that the organization can standardize the legal content of their texts in a way never before possible.

### **The Larger Political Context**

The political dimension is also a factor in understanding and determining the external conditions for tailoring and use. Here, we are particularly interested in how local tailoring is influenced by, and influences, decisions made at higher levels of the AT hierarchy (AT headquarters), as well as across the hierarchy in other branches. It is worth noting how the bottom-up process of organizational change described here took place in the context of decentralization from the top. AT headquarters closed down the group responsible for computer support for all the branches, planning eventually to create a smaller department responsible for coordinating technology development activities at the local branches. Our Århus branch was in effect an experiment in technology policy delegation. Though in line with the organization's moves towards decentralization (Markussen, in press), the delegated work and decision-making remains under strong centralized financial control.

The tailors we worked with in Århus pointed out the danger of not following through on the policy of decentralized tailoring. They sensed a lack of interest and/or resources in headquarters for following developments in the branches and saw this as decreasing the chances that lessons learned at Århus would spread to other branches. This leaves our nascent community of practice rather isolated. Because of the geographical distance between branches and their independent nature, opportunities for cross-fertilization are less likely to arise of their own accord. Instead, they need encouragement from the top.

At the same time that the tailors showed interest in spreading experience through the larger organization, they also wished to be left alone and even feared that headquarters might abruptly end the experiment and take back the PCs. The tendency at the AT is to distribute what were centrally-located technology responsibilities through the organization by encouraging (or at least allowing) bottom-up systematization. In order for this to be successful across branches, however, new forms of cooperation need to be developed that overcome the problems of geographical distance and branch self-sufficiency.

### **CONCLUSIONS**

Tailoring at AT is a process that starts from existing standard software, already capable of doing "the basics" for the organization. We have tried to show how this tailoring is a collective process, reshaping standard technologies in use to create a local, shared technical environment. The tailors we worked with at the AT cooperate to create and maintain technological and organizational structures that support, bound, and structure work practice. In order to effectively share the products of tailoring, the AT's officially appointed local developers have moved from personal solutions toward more robust, stable artifacts. At the same time, they find themselves defining, imposing and enforcing

standards. Though this can lead to a locally-defined consistency across the organization, managing the new standards makes administrative and bureaucratic demands on the tailors and the organization. In the remainder of the paper, we discuss a few of these.

Local developers at the AT are inevitably border persons [2]; they work in the fields they have chosen and for which they've been trained, as well as as technology "developers," for most a new, unfamiliar undertaking. As described by Anzaldúa, such a border life is constrained as well as enabled, often in contradictory ways, by each of the involved communities.

Crucial to the emergent community of practice of the AT tailors is their embedding in the collective work practice of labor inspection. This enables local developers to develop solutions that actually work (assuming they succeed technically). Although they are amazingly responsive to the needs and wishes of fellow workers, for practical and technical reasons they cannot satisfy everyone. The Århus branch of the AT has in part responded to this dilemma by organizationally structuring the tailoring process. For example, forums like the Technology Committee try to address conflicting opinions and demands.

### **The Tailors' Long-Term Commitment**

Sometimes, we behave as though the installation of computer technology in organizations happens almost instantaneously. In contrast, implementing (or "installing") WP/Windows at the AT was and continues to be a process stretching out in time, placing demands on the technology and on an organization in transition. Ideally, the technology should be functioning during all stages of the process (including that of the technology's own transformation), and the technological and organizational transitions should be smooth.

### **The Tailors' Techno-Social Isolation**

One of the local developers at the AT asked for a "sparring partner," someone who could bring to the discussion a more thorough understanding of technical constraints and possibilities. Frequently during the project, one of our group (Preben Mogensen), has taken on that role. A sparring partner may continue to be beneficial for tailoring at the AT, provided that this expert, recognizing the importance of the AT tailors' knowledge of local practice, engage in truly bidirectional dialogue.

The tailoring community at Århus/AT also finds itself isolated technologically both from headquarters and from potential fellow tailors in other branch offices (and more generally in other WP installations). We propose WP support groups set up by the suppliers, and broader technology support groups set up by the directorate in order to help the tailoring process survive locally at the AT office.

### The Need for Abstract Systems Thinking

"Real" system developers are encouraged to start with analysis and design; that is, to "step back" and think abstractly, organizing their work in a top-down fashion before constructing an artifact. Tailoring, however, is based neither on abstract models of use, nor on formal conceptions of the technical artifact. In the terminology of Peter Naur [22], the tailors do not have nor do they tend to develop theories about the computer application and its (possible) extensions. Of necessity, the focus of tailoring activity is on the relevant work practice, and the learning is driven by this focus. The lack of ability or training to step back and think in a more overall and abstract way may in the long run lead to problems for the tailoring process. As the number of situated solutions grows, so does the likelihood of incommensurabilities. Building more generic solutions, however, may require competencies and resources that the tailors currently lack.

### The Need for Tailoring Environments

Where in traditional systems development the processes of design/implementation and use are separate, the AT experience shows how tailors as a community of practice sit between these two processes, managing to deal with the technology from the point-of-view of the organization, and vice versa. To support this emergent community we advocate developing tailoring environments that encapsulate something of what system developers have learned over the years. These environments should support the tailors' need to move toward abstraction and systematization, while not sacrificing the benefits of being firmly grounded in the particulars of each situation. We imagine that such a tool would present a consistent use model [7] of, say, WordPerfect as a tailoring environment, and in this way point toward new possibilities of use. How exactly the model could help tailors form theories about WordPerfect is an open question that remains to be explored. (For ideas and examples of first steps toward advanced tailoring environments, see [13, 15, 21: Chapter 7].)

### Learning from Tailors?

Certain forms of system development have much in common with the kind of tailoring described here. For example, certain rapid prototyping approaches presume close connections to particular situations of use (e.g., [25]). It is our belief that the practitioners of prototyping and other forms of user-centered design can learn from the deeply embedded nature of the "system development" done by tailors. Furthermore, cooperation with systems developers could be a way for local developers to learn more about the processes of design and abstraction as well as tailoring techniques. We believe that cooperation between local developers and systems developers offers a valuable and challenging opportunity for the years to come; both sides have much to gain, and much to learn.

### ACKNOWLEDGMENTS

Thanks to the AT project participants, in particular our four protagonists for putting up with us, and showing an interest in our research; to our fellow members of the AT project group (Ellen Christiansen, Pelle Ehn, Randi Markussen, Preben Mogensen); to Soudabeh Goudarzi and Pia Lund who helped set up the standard forms as part of their studies of the use of WordPerfect; and to Lucy Suchman, Susan Newman, Leigh Star, Morten Kyng, and anonymous reviewers for their comments on previous drafts.

### REFERENCES

1. Andersen, N.E., Kensing, F., Lundin, J., Mathiassen, L., Munk-Madsen, L., Rasbech, M., and Sørsgaard, P. *Professional Systems Development: Experience, ideas and action*. Prentice Hall, New York, 1990.
2. Anzaldua, G. *Borderlands - La Frontera*. San Francisco, Aunt Lute Books, 1987.
3. Bødker, S. *Through the Interface--A Human Activity Approach to User Interface Design*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.
4. Bødker, S. Technology as a vehicle for organisational learning and change. First Socio-Cultural Research Conference (Madrid). (DAIMI No. PB-425). Computer Science Department, Aarhus University, Denmark, 1992.
5. Bødker, S. Historical analysis and conflicting perspectives--contextualizing HCI. In L. Bass, J. Gornostaev, and C. Unger (Eds.), *Proceedings of EWHCI '93, vol. 1.*, 1993, pp. 132-142.
6. Bødker, S., Christiansen, E., Ehn, P., Markussen, R., Mogensen, P., and Trigg, R. The AT-Project: Practical research in cooperative design. (DAIMI No. PB-454). Computer Science Department, Aarhus University, Denmark, 1993.
7. Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M., and Sundblad, Y. A UTOPIAN experience: On design of powerful computer-based tools for skilled graphical workers. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and Democracy--A Scandinavian Challenge* (pp. 251-278). Avebury, Aldershot, England, 1987.
8. Brown, J.S., and Duguid, P. Borderline issues: Social and material aspects of design. *Human-Computer Interaction*, 1994, in press.
9. Engstrom, Y. Interactive Expertise: Studies in Distributed Working Intelligence. (Research Bulletin No. 83). Department of Education, Helsinki University, 1992.
10. Gantt, M., and Nardi, B.A. Gardeners and gurus: Patterns of cooperation among CAD users. In

- Proceedings CHI '92* (May 3-7, Monterey, CA), ACM, 1992, pp. 107-117.
11. Greenbaum, J., and Kyng, M., Eds. *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.
  12. Hughes, J.A., and King, V. Paperwork. COMIC Report, University of Lancaster, 1993.
  13. Kiczales, G., and Lamping, J. Issues in the design and specification of class libraries. In *Proceedings of OOPSLA'92* (1992).
  14. Kyng, M. Designing for cooperation: Cooperating in design. *Communications of the ACM* 34, 12 (1991), 64-73.
  15. Kyng, M. Making representations work. In L. Suchman (Ed.), *Representations of Work*. (Honolulu, Hawaii). Hawaii International Conference on System Sciences, 1994, pp. 19-35.
  16. Lave, J. and Wenger, E. *Situated learning: Legitimate peripheral participation*. Cambridge University Press, Cambridge, 1991.
  17. Mackay, W.E. Patterns of sharing customizable software. In *ACM CSCW'90 Conference on Computer-Supported Cooperative Work*. (Portland, Oregon). ACM Press, 1990.
  18. MacLean, A., Carter, K., Lovstrand, L. and Moran, T. User-tailorable systems: Pressing the issues with buttons. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*. 1990, pp. 175-182.
  19. Markussen, R. A historical perspective on work practices and technology. In P.B. Anderson, B. Holmqvist, and J.F. Jensen (Eds.), *The Computer as a Medium*. Cambridge University Press, Cambridge, UK, 1994 (in press).
  20. Nardi, B., and Miller, J. Twinkling lights and nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies* 34 (1991), 161-184.
  21. Nardi, B.A. *A Small Matter of Programming: Perspectives on End User Computing*. The MIT Press, Cambridge, Massachusetts, 1993.
  22. Naur, P. Programming as theory building. In *Computing: A Human Activity*. ACM Press, 1992, 37-48.
  23. Okamura, K., Fujimoto, M., Orlikowski, W.J., and Yates, J. Helping CSCW applications succeed: The role of mediators in the context of use. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*. 1994.
  24. Suchman, L.A. *Plans and Situated Actions*. Cambridge University Press, Cambridge, 1987.
  25. Trigg, R.H., Bødker, S., and Grønbaek, K. Open-ended interaction in cooperative prototyping: A video-based analysis. *Scandinavian Journal of Information Systems* 3 (1991), 63-86.
  26. Trigg, R.H., Moran, T.P., and Halasz, F.G. Adaptability and tailorability in NoteCards. In *Proceedings of IFIP INTERACT'87: Human-Computer Interaction*. 1987, pp. 723-728.