In: Anzai, Y.; Ogawa, K.; Mori, H. (eds): Symbiosis of Human and Artifact, Proceedings of HCI ´95, Vol. 20B, Elsevier, Amsterdam 1995, pp. 995-1000

From Taylorism to Tailorability

Supporting organizations with tailorable software and object orientation

Helge Kahler
Institute for Computer Science III, University of Bonn
Roemerstrasse 164, D-53117 Bonn, Germany
Tel: (+49)0228/550-221, e-mail: kahler@cs.bonn.edu

Abstract

With markets globalizing and customers´ demands specializing organizations worldwide need to change. To reach the necessary flexibility of information technology one approach is tailorability, i. e. users are enabled to adjust software to their needs. Some examples for tailorability are given, and its potential benefits and shortcomings are discussed. Software development plays an important role for establishing tailorability, and object oriented methods can be helpful in this context.

1. Intro

The former successful tayloristic work model that divided labour to increase efficiency is now a big obstacle for modern business with its need to react quickly to environmental changes. The conviction that there is one best way for an organization to run and that organizations work like machines which follow the linear principles of cause and result is replaced by the idea of an organization as a constantly moving social network that keeps adapting to environmental changes (Paetau 1994). Some of the newer literature on organization introduced new concepts to overcome the rigidity of taylorism and paid tribute to a less linear understanding of organizations. The suggestions include "reengineering the corporation" (Hammer & Champy 1994) and building "virtual" (Davidow & Malone 1993), "fractal" (Warnecke 1993), or "object oriented" (Klotz 1993) organizations. All of them differ in their main focus, but all stress the importance of information technology while skipping its particular role and shape in a post-tayloristic setting.

That this must not be neglected in any organization we learned in a project finished recently in which we developed software ergonomical design principles for groupware (cf. (Herrmann et al. 1993)). The principles were influenced by interviews we conducted and that supported our hypothesis that the usage of networks which are usually designed for communication and co-operation also raises conflicts of interests between different users. An example for this is the ambiguity of visibility in a network of cooperating people: Whereas it can be of importance to one person to see what other people in a team work on it might be felt to be unwanted control by the latter. In a situation like this no single optimal solution will be available. Instead a common stable solution for all participants must be found. Moreover, the system should be able to handle ad-hoc-negotiations about conflicts by means of a negotiation function (cf. (Wulf 1993)).

In order to reach the organization al flexibility demanded by newer organizational theory as well as by the software ergonomical need to adjust systems to individuals and groups several measures and concepts are promising. One of them is *tailorability* .

2. Tailorability

Software for a modern organization should be tailorable. This means that users can adjust it to their special needs by themselves. Ideally there will be different levels of adjustment for different needs and qualifications (cf. (Henderson & Kyng 1991)). Thus, adjusting the software might mean that a person places icons or a toolbar on the screen wherever she or he wants, that the input device can be chosen (keyboard, mouse, voice etc.), but also that people make highly sophisticated configurations in a system to support their work as a team e. g. by defining who´s substituting whom at which occasion in a workflow process. Also end user programming can be used to tailor software.

Relevant dimensions of the tailoring process are who the initiatiator, the actor, and the affected persons are, what its subject is (user, task, organizational context), what its goal is, or when, for how long, and for what parts of the system it is made (cf. (Haaks 1992)).

Some examples of collaborative work practice can highlight different aspects of tailoring software in a collaborative work setting.

- When introducing a new ISDN telephony system in an organization several configurations can be made. Some of them concern technical aspects, others may result from the organizational structure and again others relate to privacy aspects. Thus, one part of the configuration adopts the telephony software to the standards of the local telephone company while another part sets the rules for call forwarding or prevents any unauthorized person to activate the microphone of one of the telephones from a remote place. Usually these configurations affect all of the people using the telephone system, and they are rarely changed. They are made *before the first use of the system*. The telephone users are seldom asked to participate in the configuration process or even informed. This is a classical and rather rigid form of tailoring.
- With SHARE Greenberg (1991) proposes a layered architecture for a conference tool with the possibility to use one of a set of "personizable floor control policies", i. e. rules for deciding who´s turn it is next to speak/write in a computer mediated conference. Such a rule could state that every person that wants to say something can just "grab" the turn by interrupting the speaking person, or that the speaking person has to explicitly stop speaking before anybody else can start. Also, there could be a chairperson who picks the next speaker from the group who raise their hands (shown by an icon on the screen). Other floor control policies can be defined. The decision for a special policy is made *before every single session*. It remains open if only the person initiating a meeting or the whole group can decide about a meeting´s floor control policy.
- A closer look at organizational aspects of tailoring is taken in some papers dealing with the sharing of customization files (Mackay 1990; Nardi 1993 Chapter 6; Trigg & Bødker 1994). While the software that is dealt with is not necessarily groupware, i. e. used by people to collaborate, there are still interesting observations about collaborative work practice made. The main point of interest here is that people individualize software they use for their daily work and share these individualized custom files with others who feel that the files are useful for them. The customization may include some macros or lengthy blocks of standard text that different people need to write to fulfill legal or other requirements. To have others benefit from the customization files "translators" (Mackay

1990) are needed who make custom files accessible and talk to people about them, thus "translating" between the system developers and the end users. They should be domain experts as well as interested in computers and willing and able to communicate and help people. The translators often emerge from a group in a "natural" way. They can be beneficial for an organization in supporting the process of tailoring software to an organization´s needs. Some positive experiences with translators ("gardeners", as Nardi (1993) calls people who are rewarded or paid for being translators) have been made. They have shown that gardeners can be a source of high quality support. Other field studies have mentioned that the process of sharing custom files has become more systematized in the course of the time (Trigg & Bødker 1994).

- Malone et al. (1992) describe a system that allows end users to tailor software on a level closer to system development than just setting parameters. Their Oval System is a "radically tailorable tool for cooperative work" where "radically tailorable" means that the tool is meant to enable end users to create different applications by modifying a working system. This is done by combining and modifying *objects*, *views*, *agents*, and *links* which provide an elementary tailoring language. While, in fact, the idea of end users designing the application that suits them best is intriguing, the question remains how many users will be able to handle the more advanced features of this complex system.

A very important aspect of tailoring in a work group setting has not yet been widely discussed. None of the examples mentioned involved a group of users jointly tailoring their groupware system to the group´s needs. This will be of increasing importance in the future since more and more groupware systems will be installed in post-tayloristic organizations. Having a group tailor their groupware system raises questions about how this can be done. Similar to the technical (cf. (Wulf 1995)) and organizational means to deal with conflicts among groupware users while they use the system, ways have to be found to moderate different interests in a groupware tailoring process.

The tailoring examples above highlight some of the potential benefits and shortcomings of tailoring software. Generally tailorable software can enhance the flexibility of an organization by enabling technical adjustments to organizational needs. Thus, for an organization tailoring can be a vital part of the management of change and the heading towards a learning organization. By actively supporting the tailoring process e. g. by gardeners spreading custom files or by work groups discussing group-relevant aspects of tailoring the self-organizational potential of work groups in a post-tayloristic organization can be set free. Also organizational support by gardeners or regular tailoring meetings can be helpful to provide structure to the tailoring process and thus keeping the organization´s technical infrastructure from turning into a thicket of incompatible individual solutions or being lost in space between system planets tailored by groups.

On the other hand tailoring software is not inevitably beneficial. There is the danger of a tailoring overhead since the time and effort needed for tailoring is lost - at least at a first glance - for the primary work task. Gardeners might be difficult to find, since they need a double qualification finding themselves on the border of system development and everyday work. Also, there is a danger of getting too unfamiliar with the work settings when a gardener works for a longer time on the developer´s side of the border. Moreover, the gardener system provides another layer between system developers and end users. While this is intended to improve the communication between these groups it might just increase the organization´s hierarchical overhead. Organizations face the danger of running into a qualification problem not only for gardeners but for all personnel. Tailoring software requires technical and social

skills, particularly in a collaborative work environment, that not every organization member might have or be willing or able to acquire. Moreover, software ergonomical design principles particularly for groupware and work psychological demands are in danger of being disrespected by unskilled tailorers. This is a dilemma software ergonomy faces by pleading for tailorability to ensure that a system fits the users´ needs on one side while on the other side struggling with the potential negative effects of letting the final look an d functionality of a system slip out of the hands of developers and putting it into the responsibility of people tailoring within an organization. Finally, allowing for more internal dynamics as a result of continuously tailoring software might be a problem for some managers fearing to lose control.

Deciding on how a system needs to be to fulfill the expectations of all or most of the people working with it at least to a certain extent doesn´t get easier when the decision is made within an organization rather than by external system developers. But the chances are better to find a way of tailoring the system to support the organization and its members well by discussing the special needs and trying out what works good and what doesn´t.

3. Software Development

Tailorability is a feature of software. Therefore, software development is a relevant area to look at. Moreover, software development is affected by the post-tayloristic organization models directly. Developing software for a special organization must no longer be document driven and follow the rigid top-down waterfall model, i. e. consist of predefined disjunct phases with written milestones to document the state of a project before the next phase may be started.

To overcome the waterfall model several approaches have been made. Two of the most promising are EOS (Evolutionary Object Oriented System Development) (Hesse & Weltz 1994) and STEPS (Software Technology for Evolutionary Participative System Development) (for an introduction cf. (Floyd et al. 1989)). Both consider software development to be evolutionary and stress the necessity to continuously coordinate the areas of development and usage since system development always includes the design of workplaces.

The particular strength of EOS is the usage of object orientation (see chapter below) which might be a good *technical* basis to realize tailorable systems. STEPS relies on user participation where developers and users exchange their views of the system-to-be or the last revision to ensure that the software fits the users´ current and future needs. Thus, part of the tailoring before the continuous usage of the software is done by system developers together with end users. This procedure for the two groups to work together in evolutionary development seems to be a good *social* approach to tailoring.

4. Remarks on Object Orientation

There are different relations between the computers science´s concept of object orientation and post-tayloristic organizations. Klotz (1993) has used the object oriented image of independently operating objects communicating through clearly defined interfaces and being provided with all necessary resources to describe how a modern organization should work and called it "object oriented organization". It is yet an open question if this is not overstretching the comparison with the object oriented approach.

But still object orientation can be of great value in designing and using tailorable software. Since people tend to think in objects rather than functions the interaction in a participatory design process between developers and users on what the work and the software are about is made a lot easier with the gap between their different notions narrowing. On the other hand

object orientation supports an evolutionary process by allowing easy prototyping and changing of modules.

Basic to object orientation are modularity by data encapsulation, inheritance of object features, and the polymorphism that frees objects that send a message from knowing the receiving object´s properties. Each of them can be very helpful for tailorability, e. g.:

- encapsulation ensures lean and clearly defined interfaces, so parts of the software can relatively easy be changed, removed, or added without risking a decrease of system stability;
- inheritance particularly supports working in a group where features of a configuration object of the work group can be passed on to the group members´ configuration objects without them having to configure each and every of their work environments manually whenever the group configuration changes;
- polymorphism is needed in tailored work settings to ensure that changing one object doesn´t make it necessary to change the methods of other objects.

These features have shown to be valuable for prototyping and will be of use to build tailorable software since many of the requirements for prototyping and tailorability are common. Moreover, for tailorable software a layered architecture seems to be useful, where each layer is responsible for one aspect of tailorability, e. g. the user interface or building and using macros. To keep these layers independent and stable to the frequent changes resulting from tailoring object orientation can provide the methods.

5. Extro

Tailorability can be part of the solution to the problem s that organizations face in a fast changing market. On the other hand it imposes more work on an organization since it takes part of the responsibility for the adequate realization of the software from the system developers. After all, building tailorable software and working with it is a process of deregulation with the chances and dangers depending on an organization´s ability to use the potential of tailorability.

More work must be directed on the tailoring practice for different settings, including the development of gardening models, concepts for tailoring as a group process, and the evaluation of tailoring activities. Moreover, technical advancement towards architectures for tailorable software must be made. Object oriented concepts might be helpful here. Finally, it is crucial to understand that organizational and technical flexibility and change are intertwined and must be dealt with jointly (cf. (Rohde & Wulf 1995) for suggestions for an integrated organization and technology development).

References

Davidow, W. H.; Malone, M. S.: Das virtuelle Unternehmen. Der Kunde als Koproduzent. Frankfurt am Main und New York 1993. Campus

Floyd, C.; Reisin, F.-M.; Schmidt, G.: STEPS to Software Development with Users. In: Ghezzi, C.; McDermid, J. A. (eds.): ESEC'89 - 2nd European Software Engineering Conference. pp. 48 - 64

Greenberg, S.: Personizable groupware: Accomodating individual roles and group differences. In: Proceedings of the Second European Conference on Computer Supported Cooperative Work 1991. pp. 17-32

Haaks, D.: Anpaßbare Informationssysteme - Auf dem Weg zu aufgaben- und benutzerorientierter Systemgestaltung und Funktionalität. Göttingen und Stuttgart 1992. Verlag für Angewandte Psychologie

Hammer, M.; Champy, J.: Reengineering the Corporation - a Manifesto for Business Revolution. New York 1994. Harper Business

Henderson, A.; Kyng, M.: There's No Place Like Home: Continuing Design in Use. In: Greenbaum, J.; Kyng, M. (eds.): Design at Work. Cooperative Design of Computer Systems. Hillsdale, NJ 1991. pp. 219-240

Herrmann, T.; Wulf, V.; Hartmann, A.: Requirements for a Human-centered Design of Groupware. In: Proceedings of the 12th Interdisciplinary Workshop on Informatics and Psychology.

Hesse, W.; Weltz, F.: Projektmanagement für evolutionäre Software-Entwicklung. In: Information Management 3/1994. pp. 20-32

Klotz, U.: Vom Taylorismus zur Objektorientierung. In: Scharfenberg, H. (ed.): Strukturwandel in Management und Organisation. Baden-Baden 1993. FBO-Verlag. pp. 161-199

Mackay, W. E.: Patterns of Sharing Customizable Software. In: Proceedings of the CSCW ´90. pp. 209-221

Malone, T. W.; Lai, K.-Y.; Fry, C.: Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. In: CSCW ´92 Proceedings. pp. 289-297

Nardi, B. M.: A Small Matter of Programming. Cambridge, Massachusetts 1993. MIT Press

Paetau, M.: Configurative Technology: Adaption to Social Systems Dynamism. In: Oppermann, R. (ed.): Adaptive User Support - Ergonomic Design of Manually and Automatically Adaptable Software. Hillsdale, N.J. 1994. Lawrence Earlbaum. pp. 172-207

Rohde, M.; Wulf, V.: Introducing a Telecooperative CAD-System - The Concept of Integrated Organization and Technology Development. In: In: Proceedings of the International Conference on Human Computer Interaction (HCI) 1995

Trigg, R.; Bødker, S.: From Implementation to Design: Tailoring and the Emergence of Systematization in CSCW. In: Proceedings of the CSCW ´94. pp. 45-54

Warnecke, H.-J.: Revolution der Unternehmenskultur. Berlin 1993. Springer

Wulf, V.: Negotiability: A Metafunction to Support Personable Groupware. In: Proceedings of the International Conference on Human Computer Interaction (HCI) 1993, 19B. pp. 985-990

Wulf, V.: Metafunctions in a Groupware: A Technical Support to Handle Conflicts among Users. In: Proceedings of the International Conference on Human Computer Interaction (HCI) 1995