Designing Tailorable Groupware for the Healthcare Domain

Robert Slagter and Margit Biemans

Telematica Instituut, P.O. Box 589
7500 AN Enschede, The Netherlands
{robert.slagter,margit.biemans}@telin.nl

Abstract. In this paper we present a theory-based approach to designing tailorable groupware for the healthcare domain. Both literature and empirical data show the need, and difficulty, of designing groupware that is adaptable to match the dynamic requirements of real-life co-operation in telemedicine. We apply an existing social theory, the Information Foraging Theory, to explain natural tailoring behaviour and state the implications for groupware design. To improve the usability of tailoring, we integrate this theoretical foundation with a groupware design approach to compose groupware behaviour out of individual building blocks and apply this to the healthcare domain. The results are a conceptual architecture and design guidelines that help groupware developers create tailorable groupware. An important contribution of our research is the concept of *task-oriented groupware patches* that help co-operating healthcare professionals in selecting, combining and fine-tuning those groupware services that fit their evolving needs and requirements.

Keywords: groupware design, tailoring, social theory, architecture, healthcare

1 Introduction

As generally stated in literature on computer supported co-operative work (CSCW) fixed technology support fails to capture the dynamics of real-life co-operation. The support provided by a groupware application has to match the tasks of the co-operating people [14;31]. However, there seems to be a real and inherent gap between social requirements and what is technically feasible [1]. In this paper we present our contribution to reduce this gap, and apply it to the healthcare domain.

In groupware, a mismatch between the task and corresponding technology support affects the co-operating people. Tailoring, i.e. adaptations to the technology support by the end-users themselves, is generally regarded as a suitable means to solve this problem [14]. However, empirical data from a longitudinal study in the healthcare domain reveals that tailorable options are currently not used to their full potential [13]. Interviews with healthcare professionals suggest that they perceive technology support in terms of activities (e.g., “transferring patient information” or “setting up a treatment plan”) [18]. Based upon these findings we conclude that there is a gap between the way people naturally perceive their co-operative tasks and the way tailorable options are currently presented.

J. Favela and D. Decouchant (Eds.): CRIWG 2003, LNCS 2806, pp. 58–73, 2003.

In this paper, we improve groupware tailoring by aligning tailoring operations with natural human tailoring behaviour and stating the implications for groupware design. We first explore how natural human tailoring behaviour can be explained by an existing social theory, the Information Foraging Theory [21]. Based upon this theory we derive the concept of task-oriented groupware patches to support tailoring.

This concept is combined with an approach to compose groupware behaviour out of individual building blocks and a detailed analysis of groupware services. This combination allows us to make well-informed design decisions for collaborative applications in the healthcare domain. The results of our research are expressed in a conceptual groupware architecture and corresponding design guidelines. The research described in this paper is based upon our empirical findings in the healthcare domain. Currently, we are applying the concept of task-oriented groupware patches to support co-operation between healthcare professionals in a stroke service.

1.1 Tailorable Groupware in Medical Settings

Tailoring is about modifying a groupware application to match changing task needs, group needs, personal preferences, changing work settings, different modes of collaboration et cetera. It is generally regarded as a key property of groupware [4;14;26;30]. Tailoring is also an important capacity of groupware systems, since many different users will have to use groupware before it reaches a critical mass [16], and because co-operative activities provide for very dynamic and diversified requirements [26]. However, tailoring is currently not used to its full potential in groupware settings (e.g., [14;20]). This also applies to the healthcare domain. Work pressure is so high that physicians strictly focus on their core task [28], and consequently do not reflect on their technology use, and thus do not tailor [13]. With these results in mind, we want to improve tailoring in the healthcare domain by designing groupware in such a way that tailoring is closely related to normal use, and does not require additional effort from the physicians. We base this design on a social theory, which guides both the description and prescription of tailoring. As in literature no single behavioural tailoring theory is available yet, we have selected an appropriate social theory from another domain and applied it to tailoring.

2 Theoretical Foundation

In order to select an appropriate theory for tailoring, the main characteristics of typical telemedicine tasks are identified. Hettinga concludes that since physicians focus on their main task and do not reflect on their technology use, they do not tailor [13]. By making tailoring part of normal system use, we expect to overcome this problem and promote tailoring. When taking a closer look at tailoring operations, two classes can be distinguished; the selection of appropriate means to support specific tasks, and the fine-tuning and shaping of those means. This is complementary to an evolutionary perspective on technology use; tailoring by selection and tailoring by variation [19]. Tailoring by selection means that for a specific task appropriate functionality and resources are integrated, tailoring by variation means that variability in (new) user behaviour is supported by for example allowing flexibility of component integration.

Based upon the characteristics of telemedicine (embedded task and selection & variation), the concept of patch-based behaviour, as part of the Information Foraging Theory [21] is selected to be an appropriate theory to both describe and prescribe tailoring behaviour.

2.1 Information Foraging Theory and Tailorable Groupware

The Information Foraging Theory (IFT) links to Rational Choice Theories (e.g., [29]) and is based upon an evolutionary ecological perspective. IFT assumes that people, when possible, will modify their strategies or the structure of the environment to increase their rate of gaining valuable information [21]. We focus upon the modification of the user environment (i.e., tailoring) to achieve task-goals. IFT draws heavily upon models and techniques developed in *optimal foraging theory* [25], which seeks to explain adaptations of organism structure and behaviour to the environmental problems and constraints of foraging for food. Although optimisation models are used in IFT, its starting point is that humans exhibit bounded rationality or make choices based on *satisficing* (cf. [23]).

Patch models in optimal foraging theory concern situations in which the environment of some particular animal has a ‘patchy’ structure. The forager has two types of actions; *between-patches* and *within-patches*. IFT assumes that information foragers allocate their time to between-patch vs. within-patch activities to achieve their goals more efficiently, effectively, and satisfactory. They typically shape the environment to fit the available strategies, and perform tasks more efficiently and in a satisficing way.

In IFT, two kinds of tailoring activities are distinguished; (1) to reduce the average costs of getting from one patch to another (*between-patches*), and (2) making more coherent patches of relevant information (*within-patches*). By analogy, between-patch behaviour provides opportunities for tailoring by selection, as the task-patch is composed of relevant media support. Within-patch behaviour is comparable to tailoring by variation as the patch can be shaped and fine-tuned to meet (evolving) behavioural requirements.

2.2 Tailoring Operations: Action and Impact

Our focus is on synchronous groupware settings in the healthcare domain. In such settings, when one person tailors, it may also affect the other user, both intentional and unintentional. For example adding an x-ray viewer during a peer review affects all participants in the session, while changing the background colour should not affect the other users. These examples show that only the relevant tailoring operations need to be communicated to the other participants.

We consider tailoring as a system change, which leads to a change in user behaviour and of course the other way around; desired user behaviour can be achieved by system changes. Tailoring operations (*actions*) lead to system changes, and typically require changes in user behaviour (*impact*). In co-operative settings, the system changes sometimes also affect the other participants, and consequently require behavioural changes of them as well.

Based on the distinction between action and impact, we consider three types of tailoring operations; adjustments, tunings and alterations. Adjustments are local changes (activity), which do not affect other people (impact). Changing the background colour of your PC is an example of such an adjustment. Tunings are local changes (activity), which require some form of tuning between people to understand the changed situation (impact), but require no action of the other user. As one physician in a videoconference turns his image viewer to black and white (due to bandwidth restrictions), the other participants should be aware of this, but do not have to act on this. Finally, alterations are global changes that create another situation (impact), and therefore require both an action and a corresponding behavioural change of the other users. An example of an alteration is a patient who wants to show the spots on his skin and therefore adds a web-cam to the audio conference, of course the physician also has to add a viewer to see the image. This categorisation of tailoring operations is summarised in Table 1.

Table 1. Action and impact based tailoring categorisation: Adjustments, Tunings and Alterations. The plus sign indicates that the operation requires an action by that person, or has an impact on that person

	Action		Impact	
	Initiator	Other(s)	Initiator	Other(s)
Adjustments	+	-	+	-
Tunings	+	-	+	+
Alterations	+	+	+	+

3 Groupware Services and Tailoring

The following section states how groupware can be designed to match the dynamic requirements of co-operation. We identify the services a groupware application should be able to provide, and the dynamics in using these services. The identification of groupware services is required to create a groupware design, which is flexible enough to cluster tailorable options in terms of task-oriented groupware patches.

3.1 The Task-Technology Fit

The services provided by a groupware application have to match the tasks of the co-operating people [24;31]. To achieve a proper matching of co-operative tasks and technology support (typically called the *task-technology fit*) we distinguish two means:

1. A good initial selection of groupware services by external experts, based on the characteristics of the collaboration “as is” and “to be”. This includes the characteristics of the collaborating people, the tasks they perform together, and the context in which the collaboration takes place (e.g., in the office, on the road, or at home).
2. Allowing the collaborating people *themselves* to adjust the behaviour of their groupware applications. This refers to tailoring as discussed previously in this pa-

per. Whenever changes to the collaborative task occur, or people change the way they collaborate, they may require different groupware services.

In both situations people need to be able to select, combine and adjust groupware services (i.e., the behaviour of a groupware application towards its end-users: the collaborating people). However, the people who have to perform these operations differ in the two situations described above. In the first one *experts* do the adjustments, while in the second one the *end-users* perform the adjustments themselves. In the latter situation there are typically no experts present to help the tailor. So, it is important to select a tailoring mechanism that is in line with natural tailoring behaviour. Our solution to this is to allow end-users to select the appropriate groupware services in terms of groupware patches that correspond to recognizable tasks. The following sections state how we identified groupware services and their dynamics in use.

3.2 Discovering Groupware Services

We selected essential groupware services from literature, based on our objective to discover groupware services that are meaningful and important to co-operating end-users. Ellis et al. distinguish three main categories of groupware services: communication, collaboration and co-ordination services [11]. As an alternative, the CoMeCo model by Ter Hofte [27] describes the services of a groupware application in terms of *conference management* (starting and stopping conferences, participation management, media management), *media* (corresponding to the communication and collaboration services distinguished by Ellis), and *co-ordination* services (floor control and access control services). Since the CoMeCo model lists groupware services that are meaningful and important for co-operating end-users, we have taken these services as a starting point. The services of the CoMeCo model that are outside our scope have been omitted, e.g., when they were aimed at groupware programmers or handled too specific situations, such as services to manipulate conference hierarchies.

Similarly we have looked at the groupware services distinguished by Dewan [7] and Kausar and Crowcroft [15]. Specifically, the services related to permissible participants, available roles and associated permissions have been adopted from Kausar and Crowcroft [15], while the services related awareness, and to process (workflow) management have been adopted from Dewan [7].

The groupware services related to starting co-operation (the *enabling* services in our model of groupware) originate from Edwards [8]. He distinguishes explicit and implicit mechanisms to start interaction. An example of an explicit mechanism is a person who invites another person for a co-operative session. Implicit mechanisms can for instance be based on similarities in (virtual) location, activities or simultaneous use of an information object.

3.3 Discovering Dynamics in Groupware Services

The way people co-operate, i.e., the mode of collaboration, changes over time [14]. Over time, people may need different groupware services to support them. We identified four aspects that have to be dynamic in a groupware application to suit the reality of co-operation. These four aspects are primarily based on literature:

1. Based on their goals, preferences, and context people should be able to select appropriate media to communicate [5]. The noisy environment of a factory may require text-based communication, while audio communication may be preferable for quick discussions in the office. Consequently, a groupware application should be able to support various forms of communication.
2. Similarly, different goals and domains may require sharing different types of information objects. While physicians need to share high-resolution x-ray images, architects have to work together on computer drawings of a house. A groupware application should be flexible enough to support such different types of shared information objects and associated services.
3. As commonly found in groupware literature (e.g., [9;15;27]), some types of co-operation require explicit user roles and associated access rights, e.g., in a workflow. Communication within large groups may be more efficient with a chairman who can grant the floor to specific people. On the other hand, explicit user roles and access rights may restrict communication in unwanted ways. So, a groupware application should optionally include services to enact a co-ordination policy and assign user roles to participants. Of course, the co-ordination policy itself should also be adaptable.
4. As described by Edwards [8] people apply different styles to join together into a collaborative session. Meeting styles range from explicit invitations to implicit chance encounters in a shared (virtual or physical) location. Related to this, people prefer to know who else is available for communication (presence awareness) and use this information to find appropriate occasions for communication [12]. So, groupware applications should be able to support various services to start communication, corresponding to different meeting styles.

Summarising: the following four aspects of a groupware application have to be adaptable to suit the dynamics of real-life co-operation: 1) the use of communication media, 2) the use of shared information objects, 3) the enacted co-ordination policies, and 4) the applied meeting style. These four aspects form the basis for our task-oriented groupware patches.

3.4 Towards Informed Groupware Design Decisions

It is not trivial to design a tailorable groupware application that provides a dynamic set of services to the co-operating end-users. To make well-informed design decisions the groupware architect needs to know how the various groupware services are related, and the actions and interactions that take place when a specific groupware service is used.

To obtain this information we have specified all identified groupware services using the formal modelling language AMBER [10]. The corresponding visual representation allowed us to specify the actions and interactions that take place when a groupware service is used, the information that is exchanged, and for instance the status information that is changed during the process. Moreover, the AMBER specification allowed us to define behavioural building blocks: groupware building blocks that provide coherent sets of groupware behaviour. These groupware building blocks helped us to define the conceptual groupware architecture described in the remainder of this paper, and revealed the services that have to be provided by the groupware infrastructure that connects the various building blocks. An example of an AMBER groupware service description is given below.

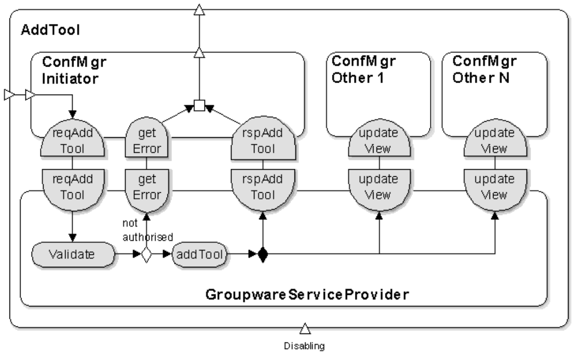


Fig. 1. AMBER description of groupware service to add a tool to an online conference

3.5 Creating Patches in the Healthcare Domain

Interviews with 13 healthcare professionals associated with the Stroke Service Enschede reveal that these healthcare professionals distinguish different types of remote co-operative activities. Most importantly, they distinguish *peer-reviews* (i.e., multidisciplinary discussions), *referring a patient*, and *requesting additional information regarding a patient* [18]. Therefore, we created three groupware patches for these physicians, related to the three types of co-operative activities. Even when on a system level the patches related to different tasks are quite similar, it is important for recognizability to present them on the user level as different patches.

The patch determines what groupware services will be activated by default (i.e., tailoring by selection) and what services can optionally be added during a conference (i.e., tailoring by variation) (table 2). Other adjustments to the groupware application can also be performed (e.g., changing fonts and colours), but are omitted in this paper for clarity. Advanced patches can also define default user roles (such as treating physician and consulted physician) and associated access rights.

Table 2. Example groupware patches in healthcare

Patch 1: Peer-review		Patch 2: Refer a patient	
Default tools:	Video conferencing Document sharing	Default tools:	Audio conferencing File transfer
Optional tools:	Audio conferencing X-ray sharing Remote microscope	Optional tools:	Video conferencing Shared scheduling
User roles:	Treating doctor Consulted doctor	User roles:	

4 Conceptual Architecture of Tailorable Groupware

This section describes a groupware design, in terms of a conceptual architecture that is flexible enough to present tailorable options in the form of groupware patches. A conceptual architecture is an organisation of computational elements and the description of their interactions [22]. To implement a groupware application according to a conceptual architecture, the described processes (i.e., groupware behaviour) have to be assigned to software components. This is not necessarily a one-on-one mapping: not every building block in the conceptual architecture will correspond to exactly one software component. However, based on the ideas presented in this paper we advocate choosing software components that provide sets of related groupware services. That way, the selection of a patch (and variations within a patch) can be directly mapped onto a composition of software components to provide the selected behaviour.

4.1 The Conceptual Architecture

Our conceptual architecture of tailorable groupware (as shown in figure 2) consists of four types of groupware building blocks, the services they provide, and the interfaces between them. The architecture reflects the identified groupware services and the dynamics in using these services.

Given our overall focus on synchronous forms of co-operation, we use the term *conference* to denote a group of people who co-operate at the same time using a particular set of communication media and shared information objects while a set of rules may apply. The *conference management* building block in our model hosts all services related to the management of such conferences. This includes services to start and stop conferences, invite people, join a conference, change the set of communication media and shared information objects and services to change the set of rules that apply. In our conceptual architecture, every valid composition of building blocks to form a groupware application contains exactly one conference block. So, independent of the selected groupware patch, these services should always be provided.

To enable communication and co-operation between participants our conceptual model contains *tool* building blocks. A tool building block provides all services related to a communication tool or shared information objects. Examples of communication tools are audio conferencing, video conferencing, and text-based chat. The services related to a communication tool allow for direct communication between the participants in a conference. Examples of tools that provide access to shared information objects are a shared x-ray viewer, application sharing and shared patient records. The services related to such tool components allow for indirect co-operation of the participants. Based on the identified dynamic aspects of groupware applications, we expect that changing the set of active tools in a conference is an important and frequently used tailoring operation. Every valid composition of groupware building blocks contains at least one tool building block.

As described by Erickson & Kellogg, people prefer to know who else is present in a shared space, and they use this awareness to start communication at suitable times [12]. Such presence awareness is provided by the *enabler* building blocks in our conceptual model. These building blocks can also provide information about ongoing conferences. A valid composition of groupware building blocks does not need to include enabler components: not every groupware application needs to provide these services.

The fourth and last type of building block, the *co-ordinator*, provides all services needed to assign roles to participants and enact a co-ordination policy accordingly. This includes the services to specify a workflow, access control or floor control. Similar to the enabler building blocks, a valid composition of groupware building blocks does not need to include a co-ordinator building block.

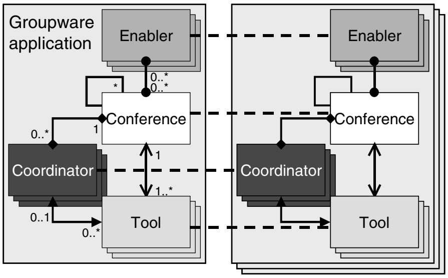


Fig. 2. Conceptual architecture of tailorable groupware

4.2 Information Exchange between Local Building Blocks

The building blocks in our conceptual architecture that form the groupware application of one participant interact to provide consistent behaviour. The conference building block for instance informs the other building blocks about the current set of participants in the conference, and the currently active tools. In a groupware implementation, this exchange of information occurs through specified component interfaces that act as contracts: by implementing a specific component interface a software component declares to be of a certain type (e.g., a tool component). Based on this information other groupware components know what to expect in terms of the services that component can offer and how to communicate with it. Such a mechanism is needed to allow for run-time changes to the groupware composition, for instance since the co-operating people decided to activate a new communication tool in a groupware patch.

4.3 Complementary Conceptual Architectures

Several other conceptual architectural models for groupware exist, such as the CoMeCo model [27], Dewan's generic architecture [6] and Clover [17]. These conceptual architectures have primarily been created to facilitate the work of groupware *designers*, the services identified in our conceptual architecture are primarily selected based on their relevance to groupware *users*.

Our conceptual architecture of groupware is an extension of the CoMeCo model by Ter Hofte. We have extended the CoMeCo model by adding services for enabling online co-operation, as identified by Edwards [8], and the presence awareness services identified by Erickson & Kellogg [12].

The generic collaborative architecture by Dewan [6] and the Clover architecture [17] are complementary to our approach. These architectures can be applied by groupware designers and programmers to design the individual groupware components in more detail when a design has been created according to our conceptual architecture.

5 An Integrated Approach: From Theory to Practice

The social-technical research presented in this paper is based on patch-based behaviour, as described in the Information Foraging Theory of Pirolli & Card [21] and our three types of tailoring operations in groupware (adjustments, tunings and alterations). These concepts are integrated with a design approach to create tailorable groupware out of behavioural building blocks and applied to the healthcare domain. As a result of this integration we derive practical guidelines for presenting tailorable options and dialog structures. This section elaborates on these results, and summarises our design guidelines.

5.1 Task-Oriented Patches and the Consequences for Groupware Design

The IFT describes individual tailoring behaviour, and states that people search for tailorable options in terms of patches, both for selecting the appropriate groupware support and fine-tuning them [21]. In groupware, a patch can be seen as a recognizable set of coherent groupware services. Two kinds of tailoring activities are distinguished; *between-patch* and *within-patch* activities. Simple heuristics guide both within- and between patch behaviour. We integrate this with the possibility to compose groupware behaviour out of building blocks that provide recognizable groupware services. As a result, we advocate creating groupware patches that correspond to concrete and recognizable tasks of co-operating people.

Our interviews [18] reveal that physicians express the services they require in terms of *tasks*. Consequently, groupware patches are named after the task they are designed to support, and consist of a selection of groupware services that are typically needed for that task. The patch states which groupware services are to be activated by default and which ones are optional (i.e., tailoring by selection and tailoring by variation). Advanced patches can also state possible user roles. Consequently, groupware patches are domain-specific. It requires knowledge of the domain and the user tasks to create patches that are in line with natural tailoring behaviour.

Obviously, tailors should be able to create new patches to cope with new collaborative tasks. Additionally, people who have specialized their groupware application by adding or fine-tuning groupware services should be able to store the new configuration as a patch. This allows users to create new patches for specialized tasks that require a specific set of groupware services (e.g., “review stroke treatment” instead of the generic “peer review”).

To allow for such selection of groupware services a groupware application should be composable of independent building blocks that provide these services. At the same time, this composition should also result in a coherent application. We have outlined a conceptual architecture that verifies these requirements.

5.2 Dialog Structures with Tailoring Impact

It is important to inform all participants of relevant tailoring operations. Dialog structures are an appropriate means to convey this information. As a conflicting force, the other participants should not be bothered with irrelevant information in dialog boxes. However, when designing groupware it is typically possible to identify whether a specific tailoring operation is an adjustment, tuning or alteration. If the operation for instance requires actions by (the application of) other participants, it is likely to be a alteration. If on the other hand, the operation only has a local impact, it is likely to be an adjustment.

Table 1 shows when dialog structures are needed to convey the impact of a tailoring operation and any required actions. This table shows that *adjustments* do not need to be communicated to the other participants, since these operations do not impact the others. *Tunings* however, need to be communicated to the other participants, while no corresponding action is required. *Alterations* not only have an impact on other participants, they even require some action by (the application of) the other participants. So in that case, the dialog structure should communicate the required action, the impact of this action, and an option to let the system immediately perform the required action.

Some alteration operations do not need additional dialog structures to clarify the impact: when an image viewer for x-rays pops up on your screen during a conference, it is obvious that one of the other participants included this tool, and no additional dialog box is needed for clarification.

5.3 Groupware Design Guidelines

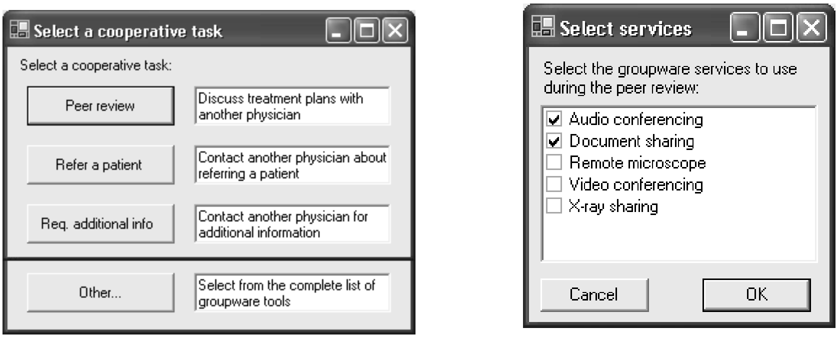
Summarising our contributions, we derive the following guidelines to design tailorable groupware:

- Create domain-specific groupware patches that correspond to a recognizable set of co-operative tasks of the prospective users;
- Allow tailors to create their own domain-specific patches;
- Upon a *tuning*: communicate the impact of the operation to the other participants in the conference;
- Upon an *alteration*: communicate the impact of the operation and the required action to the other participants in the conference;
- When designing groupware, cluster groupware services that are related to one communication tool in a separate groupware building block;
- Cluster groupware services to access shared information objects in separate groupware building blocks;
- Cluster co-ordination services (e.g., to define roles and assign access rights) in a separate groupware building block;
- Cluster conference management services (e.g., starting, stopping, joining, and leaving conferences) in a separate groupware building block;

- Cluster conference enabling services (e.g., awareness services to discover other people or other ongoing meetings) in separate groupware building blocks.

5.4 A Mock-Up Tailoring Interface for the Healthcare Domain

To illustrate our concept of task-oriented groupware patches, we created mock-up screenshots of a patch selector (see figure 3). It provides single-click access to three groupware patches. These patches represent a combination of groupware services that correspond to specific co-operative tasks (tailoring by selection), in this case: peer review, refer a patient, and a request for additional information.



Select a cooperative task

Select a cooperative task:

Peer review — Discuss treatment plans with another physician

Refer a patient — Contact another physician about referring a patient

Req. additional info — Contact another physician for additional information

Other... — Select from the complete list of groupware tools

Select services

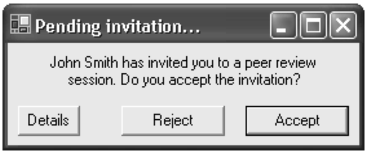
Select the groupware services to use during the peer review:

☑ Audio conferencing
☑ Document sharing
☐ Remote microscope
☐ Video conferencing
☐ X-ray sharing

Cancel OK

Fig. 3. Mock-ups of tailoring by selection and tailoring by variation

Once the peer review session has started, the physicians can still modify and fine-tune the selection of the active groupware services. This tailoring by variation is supported by the mock-up on the right. The third mock-up illustrates what the invitation may look like on the screen of the invited physician (see figure 4). Note that this invitation dialog also uses the name of the patch, as this provides the invited person awareness about the purpose of the meeting. By accepting the invitation (i.e., *impact*), the corresponding set of groupware services for the peer review session is automatically started (i.e., *action*). Details about the used groupware services can optionally be accessed.



Pending invitation...

John Smith has invited you to a peer review session. Do you accept the invitation?

Details Reject Accept

Fig. 4. Mock-up of invitation dialog (at the invited participant)

6 Discussion

We applied the Information Foraging Theory and the related patch models [21] to describe natural human tailoring behaviour, as this is in line with the two most often applied tailoring operations: tailoring by selection and tailoring by variation. Moreover, we have applied this theory to prescribe the design of tailoring in groupware by forming patches of groupware services. Our concept of task-oriented groupware patches to adapt groupware behaviour is in line with the notion of meeting *genres*, described by Antunes et al. [2].

Based on literature we have derived a set of services that groupware applications have to provide to co-operating healthcare professionals. Given our purpose to design groupware applications that can be adapted by the co-operating physicians themselves, it is important that these physicians recognise the identified groupware services. Interviews in the medical domain [18] have led us to believe this is the case.

Based on the analysed dynamics in real-life co-operation and a formal specification of groupware services we were able to make well-informed design decisions for a conceptual groupware architecture. This architecture helps developers create groupware applications that can be tailored by end-users, based on patches.

Regarding the proposed conceptual architecture: there is no such thing as an inherently good or bad architecture. Architectures are either more or less suited for some stated purpose [3]. The Software Architecture Analysis Method (SAAM) is designed to articulate those purposes and then determine the degree to which an architecture meets them. Through SAAM we verified that the groupware application has the intended properties of modifiability, composability and extensibility by tailors.

We have tested our concepts and the completeness of the specified component interfaces by creating a proof-of-concept demonstrator, the CoCoWare .NET platform. This platform consists of a groupware framework that provides generic services (e.g., security and basic services to transmit information to all participants in a conference) and a set of frequently needed groupware components. The prototype allows for runtime composition of groupware services, thus allowing us to test both tailoring by selection and tailoring by variation. For more information about this prototype, we refer to: http://www.cocoware.net

A form of validating a conceptual architecture is to show that it corresponds to implicit practices in software engineering. To do so, we studied the Groove workspace (www.groove.net) and Microsoft Windows Messenger (messenger.microsoft.com). Both products revealed an internal structure that separates tools from conference management. Microsoft Messenger also provides extensive enabling services (for discovering other people to start a meeting). Additionally, the Groove workspace identified sets of groupware tools that can be regarded as implementations of groupware patches. The Groove workspace also allows for extensibility of groupware tools and patches. Based on this process of reverse engineering we conclude that these existing applications are organised according to our conceptual architecture.

7 Conclusions

One of the fundamental challenges of CSCW is to bridge the gap between social requirements and the support offered by technology [1]. The research described in this paper aims to reduce this gap by aligning tailoring operations with natural human tailoring behaviour and designing groupware correspondingly. We have gathered requirements, and applied our findings in the healthcare domain.

In this paper, we first described individual human tailoring behaviour based on the Information Foraging Theory and the related patch models. We have integrated this theoretical foundation with a groupware design approach to compose groupware behaviour out of individual building blocks. This building block approach allows for the selection, composition, and fine-tuning of those groupware services that fit the requirements of the co-operating end-users. To be able to make well-informed design decisions we have identified recognizable groupware services, and specified them individually using the AMBER modelling language. This formal notation helped us to precisely specify the actions and interactions associated with each service, and the relations between various groupware services.

Given the obtained insights in the dynamics of real-life co-operation, we have allocated the various groupware services to behavioural building blocks. The types of building blocks, and their relations have been specified in a conceptual groupware architecture. The conceptual architecture and corresponding groupware design guidelines help groupware developers create tailorable groupware that is in line with natural human tailoring behaviour, thus improving the usability of tailoring. An important contribution of the research presented in this paper is the concept of domain-specific *task-oriented groupware patches* that help tailors in selecting, combining and fine-tuning appropriate groupware services. The component-based approach presented in this paper is needed to combine these selected groupware services into a coherent and consistent groupware application. We are currently in the process of validating our findings in various projects in the healthcare domain.

Acknowledgments. The authors like to thank Henri ter Hofte, Janine Swaak, Jan-Gerrit Schuurman, Chris Vissers and Anton Nijholt for their valuable contributions. Part of the research presented in this paper was conducted within the Dutch Freeband TeleCare project.

References

1. Ackerman, M., "The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility," *Human-Computer Interaction*, vol. 15, no. 2–3, pp. 181–205, 2000.
2. Antunes, P., Costa, C. J., and Dias, J. F., "Applying Genre Analysis to EMS Design: The example of a small accounting firm," in Borges, M. R. S., Haake, J. M., and Hoppe, H. U. (eds.) *Proceedings of the 7th International Workshop on Groupware* Los Alamitos,CA.: IEEE Computer Society, 2001, pp. 74–81.
3. Bass, L., Clements, P., and Kazman, R., *Software Architecture in Practice* Reading,MA.: Addison-Wesley, 1999.

4. Bentley, R. and Dourish, P., "Medium versus Mechanism: Supporting Collaboration through Customisation," in Marmolin, H., Sundblad, Y., and Schmidt, K. (eds.) *Proceedings of the Fourth European Conference on Computer Supported Cooperative Work - ECSCW '95* Dordrecht, The Netherlands: Kluwer, 1995, pp. 133–148.
5. Cockburn, A. and Greenberg, S., "Making contact: Getting the group communicating with groupware," *Proceedings of the ACM COCS'93 Conference on Organizational Computing Systems* Milpitas, California: ACM Press, 1993, pp. 31–41.
6. Dewan, P., "Architectures for Collaborative Applications," in Beaudouin-Lafon, M. (ed.) *Computer Supported Cooperative Work (CSCW)* 7 ed. John Wiley & Sons Ltd., 1998, pp. 169–194. Retrieved from: ftp://ftp.cs.unc.edu/pub/users/dewan/papers/arch2.ps.Z
7. Dewan, P., "An Integrated Approach to Designing and Evaluating Collaborative Applications and Infrastructures," *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 10, no. 1, pp. 75–111, 2001.
8. Edwards, W. K., "Session management for collaborative applications," *Proceedings of the 1994 ACM conference on Computer supported cooperative work* Chapel Hill, NC, United States: ACM Press, 1994, pp. 323–330.
9. Edwards, W. K., "Policies and Roles in Collaborative Applications," *ACM Conference on Computer Supported Cooperative Work (CSCW'96)* New York: ACM Press, 1996, pp. 11–20.
10. Eertink, H., Janssen, W. P. M., Oude Luttighuis, P. H. W. M., Teeuw, W. B., and Vissers, C. A., "A Business Process Design Language," in Wing, J. M., Woodcock, J., and Davies, J. (eds.) *FM'99 – Formal methods: World congress on formal methods in the development of computer systems, Toulouse, France, September 1999; Proceedings, Vol. I. Lecture notes in computer science: 1708* Heidelberg: Springer, 1999, pp. 76–95.
11. Ellis, C. A., Gibbs, S. J., and Rein, G. L., "Groupware: Some issues and experiences," *Communications of the ACM*, vol. 34, no. 1, pp. 38–58, 1991.
12. Erickson, T. and Kellogg, W., "Social translucence: an approach to designing systems that support social presence," *Transactions on Computer Human Interaction (TOCHI)*, vol. 7, no. 1, pp. 59–83, 2000.
13. Hettinga, M., "Understanding evolutionary use of groupware." PhD thesis Telematica Instituut, 2002.
14. Kahler, H., Mørch, A. I., Stiemerling, O., and Wulf, V., "Introduction to special issue "Tailorable Systems and Cooperative Work"," *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, vol. 9, no. 1, pp. 1–4, 2000.
15. Kausar, N. and Crowcroft, J., "An architecture of Conference Control Functions," *Proceedings of Photonics East* Boston, MA.: SPIE, 1999.
16. Koch, M. and Teege, G., "Support for tailoring CSCW systems: Adaptation by composition," *Proceedings of the 7th Euromicro Workshop on Parallel and Distributed Processing* IEEE Press, 1999, pp. 146–152.
17. Laurillau, Y. and Nigay, L., "Clover Architecture for Groupware," *Proceedings of the 2002 ACM conference on Computer supported cooperative work* New York, NY, USA: ACM Press, 2002, pp. 236–246.
18. Michel-Verkerke, M. B., Schuring, R. W., and Van Harten, W. H. Needs, requirements, interests and impact of integrated wireless ICT in care: TeleCare/D2.2. 2003. Enschede, The Netherlands, Twente University.
19. Mørch, A. I., "Evolutionary growth and control in user tailorable systems," in Patel, N. (ed.) *Adaptive Evolutionary Information Systems* Idea Group Publishing, 2002, pp. 30–58.
20. Oppermann, R. and Simm, H., "Adaptability: User-initiated individualization," in Oppermann, R. (ed.) *Adaptive user support. Ergonomic design of manually and automatically adaptable software.* Hillsdale: Lawrence Erlbaum Ass. Publ., 1994.
21. Pirolli, P. and Card, S., "Information Foraging," *Psychological review*, vol. 106, no. 4, pp. 643–675, 1999.
22. Shaw, M. and Garlan, D., Software Architecture: Perspectives on an Emerging Discipline Prentice Hall, 1996.

23. Simon, H. A., "Rational choice and the structure of the environment," *Psychological review*, vol. 63, no. 2, pp. 129–138, 1956.
24. Slagter, R. J., Biemans, M. C. M., and ter Hofte, G. H., "Evolution in Use of Groupware: Facilitating tailoring to the extreme," in Borges, M. R. S., Haake, J. M., and Hoppe, H. U. (eds.) *Proceedings of the 7th International Workshop on Groupware* (CRIWG2001) ed. Los Alamitos,CA.: IEEE Computer Society, 2001, pp. 68–73.
25. Stephens, D. W. and Krebs, J. R., *Foraging theory* Princeton, NJ: Princeton University Press, 1986.
26. Stiemerling, O., "Supporting tailorability of groupware through component architectures," *ECSCW'97 Workshop on Object Oriented Groupware Platforms (OOGP)* Lancaster: 1997, pp. 54–60.
27. ter Hofte, G. H., *Working apart together: Foundations for component groupware*, Enschede, the Netherlands: Telematica Instituut, 1998. Retrieved from: http://www.telin.nl/publicaties/1998/wat/wat.htm
28. Tucker, A. L., Edmondson, A. C., and Spear, S., "When problem solving prevents organizational learning," *Journal of Organizational Change Management*, vol. 15, no. 2, pp. 122–137, 2002.
29. Turner, J. H., *The structure of sociological theory*, 5 ed. Belmont, CA: Wadsworth, 1991.
30. Wulf, V. and Golombek, B., "Direct activation: A concept to encourage tailoring activities," *Behaviour and Information Technology*, vol. 20, no. 4, pp. 249–263, 2001.
31. Zigurs, I., Buckland, B. K., Connolly, J. R., and Wilson, E. V., "A test of task-technology fit theory for group support systems," *ACM SIGMIS Database*, vol. 30, no. 3–4, pp. 34–50, 1999.