Tailoring Infrastructures: Supporting Cooperative Work with Configurable Email Filters

Volkmar Pipek[1], Markus Won[2], Roman Englert[3], and Volker Wulf[4]

[1] IISI - International Institute for Socio-Informatics
Heerstr. 148, 53111 Bonn
volkmar.pipek@iisi.de
Laboratory of Human-computer Interaction and Group Technology
Department of Information Processing Science University of Oulu
Linnanmaa, FIN-90570 Oulu, Finland
volkmar.pipek@tol.oulu.fi

[3] Conet AG
Theodor-Heuss-Allee 19, 53773 Hennef, Germany
mwon@conet.de

[2] Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
roman.englert@telekom.de

[4] Institute for Information Sciences, University of Siegen
Hölderlinstr. 3, 57068 Siegen, Germany
wulf@fb5.uni-siegen.de
Fraunhofer Institute for Applied Information Technology (FhG-FIT)
Schloss Birlinghoven, 53754 Sankt Augustin, Germany
volker.wulf@fit.fraunhofer.de

Abstract. In fragmented work settings like network organizations or virtual organizations, monolithic approaches to implement support for collaboration would require the actors involved to agree on the usage of the approach or tool under consideration. As the autonomy of actors in these settings makes this hard to achieve, we suggest an exploration and an increase in the end-user tailorability of basic software infrastructures to enable even actors in theses settings to tailor their collaboration support to their needs. An example for this strategy is illustrated by using email as a basic groupware technology. We use server-based email filters to improve the coordination of work processes and increase group awareness in these settings, and focus on making it easy for end users to understand and tailor the technology according to their needs. We use and enhance concepts from the discussion on the "tailorability" of CSCW systems (a visual filter composition language, a component-based architecture and additional support for exploration and documentation) to implement and evaluate our prototype.

Introduction

Computer support for cooperative work has not only become a key success factor for organizations, but also the enabling technology for new forms of inter-organizational cooperation and distributed work. Virtual organizations (Picot et al., 1998; Mowshowitz, 1997) evolved as a new form of joining core competencies of different actors (individuals and organizations) to offer products and services beyond the skill and knowledge of the individual actors.

Comparing the technologies and tools used in virtual organizations to those in traditional forms of organizations we find heterogeneity rather than homogeneity, different tools, technologies and usages rather than a uniform, standardized groupware platform. Cooperation usually takes place in a heterogeneous infrastructure with some shared standards and basic technologies, but with many different tools and usages (cf. Rittenbruch et al. 1998, Törpel et al. 2003). Like in other collaborative settings, there still remains the necessity to be able to flexibly configure the infrastructure that is collaboratively used according to new or changed work contexts (Henderson and Kyng 1991). This characteristic of “Tailorability” has up to now mostly been explored for monolithic CSCW tools or frameworks (e.g. Malone et al. 1992, Wang and Haake 2000). But there is the opportunity as well as the need to transfer tailorability to the domain of heterogeneous software infrastructures. From a user's perspective on their collaboration infrastructure as a whole, the tailoring support should mark "Group Tailoring Hot-Spots" of the infrastructure.

As a consequence of the technological heterogeneity encountered in virtual organizations, and a lack of motivation among actors to agree on shared standards, it is often email that remains most common denominator of cooperation technology. Important groupware functionality, like awareness support of workflows or cooperation structures is then usually not available in these settings.

In our work we explore the idea of fostering group support in heterogeneous software infrastructures by using server-side email filters to implement a lightweight technological support for group collaboration in these settings. Using standard email protocols and standard web technology we try to avoid interference with the technological infrastructure of the users.

Rather than providing as much groupware functionality as possible we focused on providing end-user friendly means to tailor the functionality to the given requirements of a specific collaborative setting. Here, we applied and enhanced concepts known from earlier research on tailoring and end-user development (Liebermann et al. 2005).

Related Work

Exploring infrastructure technology for collaboration support

There are various experiences to consider in the context of our work. A number of publications have recently paid more attention to the fact that support of collaborative work does not often operate on a stand alone basis, but rather is embedded in a technological infrastructure (Dourish 1999, Dourish and Edwards 2000, Hanseth & Lundberg 2001, Dourish 2003, Pipek and Kahler 2005).

Dourish (1999, with Edwards 2001) elaborated on the problems that the 'layeredness' of infrastructures, particularly the necessity to rely on lower layers of infrastructure when constructing a tool infrastructure for collaborative work, causes in the context of the development of collaborative systems. He also presented a concept of collaboration support that intertwines with the file system of an operating system instead of providing a separate tool (Dourish 2003). Hanseth and Lundberg (2001) addressed the role of standards in software infrastructures. On the one hand, standards are necessary to provide the compatibility among software tools, but on the other hand, standards also define structures of in- and exclusion of information and resources in collaborative systems. Pipek and Kahler (2005) addressed similar issues, and discussed the problems and opportunities for tailoring support by categorizing existing approaches to support tailoring in CSCW tools.

Email technology

There are several problems for getting optimal support from this technology. There is no automatic support e.g. for different email categories (meeting request, note, reminder, etc.) except using the subject line, send mails as CCs, etc. If there is such functionality, it is usually client-based and therefore user-specific. So, some email clients (i.e. Microsoft Outlook) allow for sending meeting arrangements, or they create automatic answers. The disadvantage is that nobody knows which types of email clients are used by others and how they are configured. Even though the overall design of email clients has been discussed extensively lately (Gruen et al. 2004), there has been not much progress regarding email usage on the group level.

As described, many approaches for email-based group support rely on client-based technologies. There are server-side email configuration systems as Procmail (Hampton, 2000) but they are very difficult to use for non-programmers since they are script-based. Users need access to the server to configure such systems. To overcome this technological obstacle, web interfaces for these filter system (e.g. Websieve) have been developed. But these do not support a group-oriented approach (e.g. configurations always relate to one user). In practice, often only simple support, e.g. automatic replies in case of absence, is actually being used. Email-based tools for supporting collaboration have been developed earlier (e.g. Malone et al. 1988, Camino et al. 1998). Using filters to support group cooperation is a new idea in that context. Using an email system with enhanced capabilities (group support, etc.) has also never been explored with sufficient support for end users so far.

Support for Tailoring

Tailorability today is a well-accepted property of information and communication systems designed to support group work. To fit the changing needs of group work in organizations, the technological infrastructure (i.e. the software tools) has to be flexible, and the flexibility has to be manageable for the users. The core question in the tailoring discussion is what can be done to adapt tools and related work practice in a use context to each other to support cooperative work in an optimal way (i.e. Henderson and Kyng, 1991). It has also been discussed how these aspects change the basic design of the software artifacts which are to be tailored.

The "architectural" perspective explored tailorability to develop concepts and examples of very flexible software systems, which could be adapted to their use scenarios (MacLean et al., 1990; Malone et al. 1992). Object-Orientation and Component-Based Systems (Stiemerling and Cremers, 2000) have been explored to increase the flexibility of software artifacts designed to support group work, other approaches addressed issues of analyzing, separating and composing tailoring entities along the typical functionality of CSCW systems (Malone et al., 1992).

The "user-interface" perspective explored how tailorable software should present itself to the tailors. Henderson and Kyng (1991) addressed the question, who the tailor is, and distinguished three levels of tailoring (choosing between predefined alternatives, constructing new artifacts from existing pieces, and reprogramming the artifact) which require different levels of expertise regarding the supporting technology.

While the general discussion on tailorability was associated with CSCW settings, the discussion of these issues is now continued under the label of end-user development and considering a wider scope of tools (Lieberman et al. 2005, Sutcliffe and Mehandijev 2004).

Component-based tailorability

Component-based architectures are very common in modern software development, and allow for better reusablitity of software components. Thus, software development becomes faster (and cheaper) as well as the quality can be increased (by using tested and well-known components). Not only do developers benefit by this technique, end-users too avail themselves of it if component-based architectures are used to build highly tailorable and manageable (for end-users) applications (Morch et al. 2004). The most striking advantages are:

- The component concept is easy to understand: An application or composition consists of several components. Each of those components has its own function and they all work together as one system. The communication between components is done by sending messages them.
- The component approach allows for very powerful tailoring mechanisms: Components can be chosen, they can be parameterized and they can be bound together. Thus, the tailoring language only consists of three basic but very powerful operations.
- The visualization and the tailoring operations (visual tailoring language) can be figured out easily: Based on the first two points composing an application can be done by "drawing" it (Stiemerling, 2000).

If we regard an email filter (or a set of email filters working together) as a component-based application then we can use the same techniques as mentioned above.

Several user studies have shown that those concepts can be easily understood by end-users (Wulf, 1999). In several thinking-aloud tests (Nielsen, 1993) and different applications as well as in a field study this was evaluated (Wulf, 1999). Here we found out that the critical point is to understand the semantics of a component (a single one as well as a compound component).

Tailoring interfaces

Obviously, ordinary groupware users can not be expected to acquire programming skills to be able to tailor an artifact accordingly. Several approaches, some inspired by Nardi's (1993) work on end-user programming which aim at developing tailoring environments which provide simple concepts and interfaces for end-user (i.e. Malone et al. 1992).

There are several techniques that support using an application or learning its functionality. As mentioned above, in our case we first have to provide simple ways to understand single components. Furthermore, the workflow (or event flow) of a composition has to be taken into account.

In general, looking at features that support learning of tailoring languages we can draw on experiences concerning ordinary functions in single user applications. Tailoring environments for users without programming skills should be designed consistently with ordinary functionality (cf. MacLean et al., 1990; Nardi, 1993).

Features that encourage learning of single user applications allow structuring, describing, experimenting with and exemplifying the usage of the functionality (e.g. Caroll, 1987; Paul, 1994). These features are provided by programmers for the users.

In the following, we address several concepts mentioned above that can be used to ease the learning of how to tailor or compose own filter as well as understand "pre"-configured email-filters (i.e. filters composed and used by other users).

Experimenting and Exploration: Mackay (1990) as well as Oppermann and Simm (1994) found that experimentation plays a major role in learning tailoring functions. Nevertheless, Mackay (1990) reports that the fear to break something is a barrier to tailoring. Oppermann and Simm (1994) found that the effects resulting from experimenting with tailoring functions are difficult to perceive. "Undo function", "freezing points", "experimental data", and "neutral mode" are features which support users in carrying out experiments with a system's function. Especially in the context of email or groupware in general experimental data can lower the barrier of tailoring as testing of system changes can affect other users.

Exemplifying: Examples provided by other users are an important trigger to tailoring (Wulf and Golombek, 2001). Animation machines present a recorded sequence of interaction. Such animation gives an example on how users can apply certain functions.

Describing: Mackay (1990) found that the lack of documentation of respective functions is a barrier to tailoring. Manuals and help texts are typical means to describe the functionality of applications. A description that is provided by the software vendor informs users about the state transition within the software system in which the execution of a function results.

Pre-Study: Using Email in Virtual Organizations

Virtual organizations usually comprise of actors with a high level of autonomy that usually work in distributed setting for a limited time period to achieve a shared goal. Email technology plays an important role in connecting members of virtual organizations with each other (Picot et al. 1998). More sophisticated and more powerful groupware technologies usually are available, but only in the form of complex products that have to be bought (costly), installed (time-consuming) and administrated (both). Additionally, users have to be trained. Users in a virtual organization often are very skeptical whether the benefit associated with the use of a groupware platform is worth the investments necessary to install, maintain and learn about the groupware platform. Thus, they usually fail to agree on one technology to use (Törpel et al. 2003).

Email often remains as the only common technology to use for cooperation. In an explorative user study we further explored how email is being used as a collaboration technology in virtual teams. 22 interviewees from 7 organizations, working in distributed work settings and using heterogeneous infrastructures were asked to describe their current work contexts and how team members and external partners collaborate. Some communication scenarios were discussed regarding the specific problems that occur. We learned that usually the subject and the sender of an email were used to identify the topic or the importance of an email. They should give a first insight how server-based email filtering could ease collaboration. Finally the participants were asked to outline their own scenarios. The resulting key scenarios for email usage were:

- Meeting arrangements: Those emails often include more than two persons. Answers are sent to all recipients then. This procedure requires many emails to be sent and read.
- Replying to external inquiries: sometimes produced by an email form integrated in a web site have to be answered. Sometimes these inquiries are archived (especially the included addresses) and the answer has to be checked by a colleague.
- Exchanging documents: The most common way to exchange document is to send them by email. Whereas working documents are only passed between team members, camera-ready document have to be stored at a special place. Furthermore, at some stage the team leader has to be informed about current work progress.

Those three scenarios show how email is used to coordinate work or collaboration. In many cases copies are generated and sent to another team member (superior, secretary, archive). Often this can be done semi-automatically depending on email properties such as subject line, recipient, including attachment (i.e. name contains “final version”), etc.

Looking at different types of emails we found that,

- recipients distinguished between personal mails (having one recipient) or "information mails" (more recipients or addressed to a group email alias),
- recipients distinguished external and internal (colleagues) senders, and
- in general, most emails were sorted only by checking the subject line and the sender’s name, not by reading the mail itself.

These observations informed the design of email-based group support. On the level of collaboration, the following functions were assessed as helpful in a group scenario:
Group-related configurations: Relating filtering rules not only to a person but to a group is an important step for establishing a shared notion of email-based collaboration understanding of email usage.

Transparency of configurations: In collaborative settings it is important to understand other individual's or other group's handling of the shared technology. Occasionally transparency is also necessary to understand one self's sorting schemes in order to find an email.

Awareness support (e.g. Sandor et al. 1997): If users operate with email-based concepts to coordinate their work, a peripheral awareness of the email correspondence and the flow of work can be useful. Emails can be automatically sent to other's (interested people) as carbon copy (CC) or they can be forwarded.

To bring our ideas into practice, we estimated appropriate means for configuration and tailoring as more important than supporting more sophisticated concepts.

Concept and architecture

Similar to Dourish's (2003) way of enhancing a file system, the basic idea to introduce group functionality for collaboratively using email in fragmented work settings is to use tailoring opportunities on 'deeper' layers of the shared infrastructure. The use of server-side email filters has the advantage of implementing groupware functionality at a location that can be shared by users even in heterogeneous infrastructures. Additionally, we exploited ideas from the discussions on tailorability to guarantee the system's usability for end users.

Using a component-oriented approach

Emails can be seen as messages that are passed between sender and receivers via a chain of servers. Server-sided email filters can provide additional navigation components (and connections between them) between individual mailboxes or mail servers. As described above, traditionally filter systems are to be configured by scripting languages, which are not suitable for most end users. We decided to describe the filter system as a net of components. Aside from using component-based visualization concepts, we also used component-based technology for transforming our component-based filter descriptions into a script that can be understood by the server (and vice versa). In using components on both levels, our system remains scalable for further extensions.

Useful and manageable filter criteria

Our filter concept uses the MIME email properties (cf. MIME standard) to process emails. In addition to the MIME standard, it is possible to introduce new X-Tags (self-defined mail attributes) that can be used for filtering. The prototype presented here does not integrate all technical possibilities but aims at checking out which of the possible properties of emails and the resulting filters are relevant. Most of the scenarios we encountered in our pre-study can be automated with the help of filters that analyze the standard mail tags in various ways. The prototype should integrate the following kinds of filter techniques:

- Checking the sender's name (on equality)
- Checking the subject (subject contains one or a set of words)
- Checking the recipient list.

As described above, additional filter types can be added easily (date, time stamp, visited server, etc.). Regarding the possible actions that are being performed based on filter results, our prototype needs to be able to

- generate copies of existing mails, and
- create notification mails (about what was send or received)

Architecture

Figure 1 shows an architectural view of our system. Client and Server are not modified by our enhancements. With our component-based approach we only changed the visualization of the filters, and not their functionality. There is also a local storage for the component-based mail filters which are converted into the script-based filter files every time the filter configuration changes. The filter administration environment accesses the filter configuration storage and changes its content. It merely serves as an editor of the filter configurations that are stored on the server. The email client has access to the email server the same way as if there was no filter system.

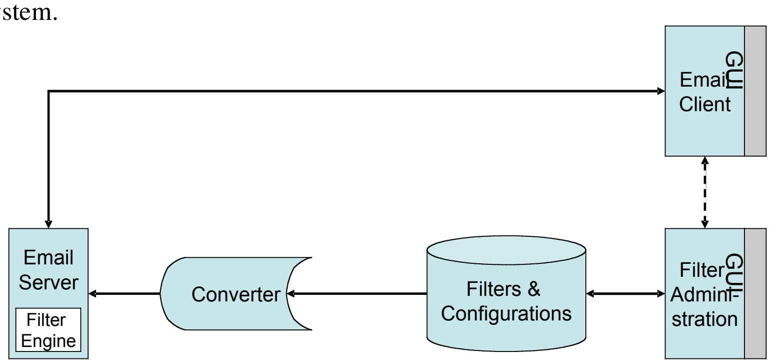

Fig. 1. Architectural Model

The tailoring client

As described above, our component-based approach not only allows for very flexible applications but is also easier to handle for non-programmers. Filters or little 'filter workflows' can be designed as a component net. In the visual tailoring language, we distinguish four kinds of components:

- Inbox (before filtering): This is where an email first arrives.
- Mailbox (after filtering): The final box the remaining emails fall into after all filters have been executed properly.
- Several conditions: Conditions (i.e. "contains subject 'Project A'?") always can be answered with a "yes" or a "no". According to the answer something happens to the mail.
- Created emails: Those are the actions of the email filter system. After a condition is true a new email can be created. This can be the original message or a new (automatically generated) message. As discussed above, the email filter system also uses email as an awareness mechanism. For example, emails are created and sent automatically if something happens that should be considered as important (depending on the filter configuration).

All those components (switches with conditions, new email, etc.) are bound together in a complex component and represent one filter configuration. At the tool's front end the filters are visualized so that changes to the composition can be done easily by adding a component (drag it from a toolbox) and connecting it (drawing lines between components).

To enable users to understand the whole system it is necessary to guarantee that all filters within the group are public and can be looked at. For this reason it is possible to also open other persons' email filters. Another reason for looking at someone else's filter configurations is to learn and understand how filters can be used. Filters can be used as examples that can be copied into the own filter configuration and customized.

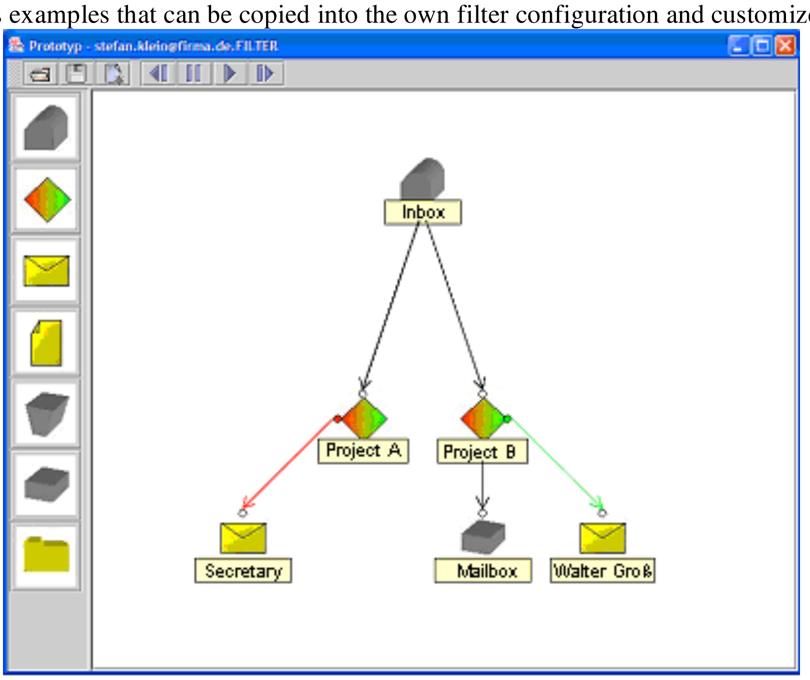

Fig. 2. Screenshot, analyzing filter configuration

Exploration mechanisms have been added to ease the tailoring for end users. Contrary to single-user applications like text processors, groupware services like email have the specific problem that they usually can not be 'tried out' because other users may be disrupted. Additionally, 'learning by doing', undo buttons, etc. are mechanisms that do not work in an email-based system since emails once sent can not be taken back.
Consequently, we had to integrate possibilities that ease learning and understanding, as well as, to provide means for testing or exploring tailored filters. Users can integrate annotations to whole filter nets as well as to parameterized components and their meanings within a filter. For the description of the exploration mode, see also the descriptions below.

Using the prototype

First, we illustrate an example of its application, and explain the aspects of our prototype. Figure 2 depicts a screenshot of the filter administration tool mentioned in figure 1. Email filters are connected to processes (or 'workflows'). Normally the inbox of the filter system is on the top of the screen and marks the beginning of the process. Arrows mark possible directions emails can take according to special conditions. In figure 2 we see a filter which first doubles incoming mails. After that "the left one" is checked if it belongs to "Project A". If not so (red), it is forwarded to the typing pool (secretary). Then the "right mail" is checked on belonging to "Project B". In this case it is forwarded to "Walter". Additionally, it is put into the mail client's inbox. Changing the configuration can be done easily by deleting the arrows or adding filter components (see toolbox on the right side) using 'drag and drop' functions. The configuration of the switches can be done by clicking on it and editing the conditions. Several filter properties are also visualized. Those conditions are very simple boolean expressions as they have to be understood and expressed by the users themselves (i.e. Recipient = walter@noorg.de or Subject CONTAINS "important"). Combinations of expressions are allowed (using AND). Furthermore, every expression can be described in own words (comment) to ease understanding.

Additional tailoring support

To provide the necessary end-user orientation we integrated three mechanisms:

- Tool tips: Explanations are given by textual tool tips to each filter element. For example, as shown in figure 2, the condition "belongs the mail to Project A" is described as "Project A?". So, inexperienced users can understand different filters and learn their functionality only by reading the annotations (some of which are generated by the system, some enhanced by the users themselves).
- Visualizing other users' filters: Other group members' filters can also be loaded into the filter administration tool, although they can not be changed. This eases the

understanding how filters can be used efficiently and how they work. For privacy reasons, of course, the emails themselves remain invisible.

- Exploration mode: Users can easily check the configured filter with a test mail. The user then can take an existing mail (or a newly generated one), drag the email from the email client, and drop it onto the inbox of the filter administration client. From this stage, the mail is passed on sequentially through the filter. Newly generated mails (notifications) are visualized the moment they enter the system. If the email is forwarded to another user and other filters are being activated, a second filter window appears. So the user (filter configurator) can test the filter settings with “real” emails without actually sending them through the mail system, and therefore without disturbing other users.

All three techniques ease the understanding of the system. More experienced users learn the functionality of the current filter settings by analyzing the graphical structure of the filter and the additional text information. More complex settings – especially several users’ filter settings that have to be taken into account for understanding the group’s joint working space can be learned by exploring the system’s functionality using emails with the according properties (tags, headers, etc.).

Learning by copy and tailor

As described above the transparency mode can be used to learn about other users’ filter settings. If such filters are displayed they can be stored as filter components and integrated into own filter settings. By using this feature less experienced users can tailor their own filter settings by simply copying and adapting existing ones.

Furthermore, local administrators or super users can use this feature to design filter components for regular collaboration scenarios within their organization.

The quality of copied filter settings here can only be assured by organizational rules. For instance, administrators may store predefined filters at a special directory. If users include some of those filter settings they know that they were designed by experts and have been tested.

Awareness and transparency functionality

Our email-based approach provides not only for awareness support, but also static transparency of others' filter configurations.

The filter techniques allow establishing awareness services during normal use of email. Notification emails can be generated and send to colleagues if email with certain properties are being sent or received. For example, emails of important customers (identified by "sender") will be doubled and notification mails will be sent to group members.

By using the exploration mode that allow for investigating of all the users’ email filter settings, work processes can be understood more easily. This is particularly important in virtual organizations that usually have no centralized organizational structure or transparent processes that could ease cooperation. Both awareness concepts allow for more transparency within the distributed organization. In the first case, the prototype was built without taking into account any privacy issues. Thus, all users within the group are allowed to browse through all email filters which leads to maximum transparency and learning effect. In the future, privacy issues have to be taken into account more seriously as filter settings may be a sensitive part of self-organization (for individuals as well as to groups).

Integration into existing infrastructures

The main goal of our approach was to allow for enhanced collaboration scenarios in virtual organizations where not all the members use the same technical infrastructure. Thus, what we needed was a technological basis that is available for all potential members without much technical overhead, and which allows greater end-user management flexibility. Email servers nowadays are very easy to handle and to administer (i.e. integration of new users, connecting to the email clients) and there are many providers who offer own email servers on their machines. Most of them are able to interpret script-based filter languages like procmail. This 'deeper' infrastructural level provides an access point for group-oriented functionality.

In the implementation of our prototype, the visualization and configuration of those filters can be done by using our filter administration client which is completely Java-based and therefore executable on many different platforms. The configuration files (XML) have to be stored in a highly accessible storage location.[1] This could be a web server for example. Simple access rights ensure that only the group members are allowed to read and change the contents. Using these technologies, that are the common denominator in today's internet-based infrastructures, it is possible to embed our concept into almost any given work infrastructure.

On the filter server all email accounts of all participating users have to be accessible as the account information is needed to transfer the visually built filter configuration to the email server.[2]

User experiences

To get some information about the feasibility of our concept in practice, we tested our prototype regarding its usability with a small number of users. Our evaluation was based on heuristic evaluation (Nielsen 1992). Three types of persons with different experiences and technical background were interviewed: "normal" users, professional users with programming skills but no administration experience, and one administrator. All of them are working in the field of IT consulting. Thus, they all have been working in virtualized and distributed working scenarios for a long time. First, they were presented with the idea of the software, and later the functionality was presented. After this step, they were asked to answer several questions

---

[1] The "filters and configurations" server is a centralized one. Instead of, different mail servers can be accessed by it.

[2] In fact, the email server then receives procmail scripts which are based on the email configuration.

concerning existing filter configuration. Finally, they were asked to configure own filters. The main results were:

- Exploration mode supports easy and exemplified learning: During the phase in which the interviewees should explain existing filter configurations the two more experienced users interpreted the annotations well, whereas the less experienced one used the exploration mode and generated different mails to understand the functionality of the filter.
- Building by exploration: In the third phase, the test persons had to configure their own filters. Both the programmer and the administrator designed their filters quickly by ‘dragging and dropping’ the filter components and binding them. The third person interactively developed parts of the filter, tested them by using the exploration mode and repeatedly changed parameters within the components. Interestingly the resulting filters differed in design but not in functionality.

The graphical tool was very helpful even for experienced users. Especially when filter configurations become more complex the graphical view is easier to understand compared to textual descriptions of filters.

Especially less experienced users felt very comfortable when using the exploration mode. It helped the understanding of the system’s behavior. The possibility to generate emails using the familiar email client and using them for testing the configuration increases the understanding not only of the filter system but also of email messaging in general.

Conclusions

The new forms of organization and collaboration like virtual or networked organizations produce an increased the level of fragmentation of work settings. To also support these settings with groupware technology, it becomes necessary to explore deeper levels of the technological infrastructure at hand, and those levels that represent the most common denominator of the technologies used by the potential collaborators. Dourish (2003) described such an approach at the level of file systems. We have now described such an approach using standard email technology to provide groupware functionality (awareness support, modeling of small workflows, additional notification services). To make these infrastructures adaptable for end users, we not only have to find "tailoring hot spots" within the shared technology and exploit them for group tailoring support, but we also have to provide the flexibility offered by the software infrastructure in a way that makes it manageable by end users. Therefore, a significant amount of our conceptual work went into the application of tailoring support concepts from standard CSCW tools for these ‘infrastructural’ technologies of email and web servers. In a small evaluation study, we were able to show the suitability of our concept for end users. Still, there are many open questions around this field of research. Following the use and the further development of email filters over time, especially in those cases where the mutual filter settings are known by all users of a group. The installation of an open community support concept around the configuration of the filters, as e.g. described in the concept of the ‘Use Discourse Environments’ by Pipek (2005) can help foster the appropriation of the technology on a social level. That way, in the long run, it would be possible to find collaboration patters based on email communication that could then be supported more specifically. Elaborating on earlier work on semi-structured messages (Malone et al. 1988), a decentralized and standardized support of message filtering could be a powerful infrastructure for the information and knowledge management of the enterprises of the future

Acknowledgements

We would like to thank Radhakrishnan Subramaniam and the anonymous CRIWG reviewers for their comments on earlier versions of the paper. The research presented here has been partially funded by the German Ministery for Education and Research within the 'Olvio' project under the reference number 01HG8890.

References

Camino, B. M., Milewski, A. E., Millen, D. R., & Smith, T. M. (1998). Replying to email with structured responses. Int. Journal on Human-Computer Studies, 48, pp. 763-776.

Caroll, J. M.: "Five Gambits for the advisory Interfacs Dilemma", in: Frese, M., Ulich, E., and Dzida, W. (eds.), Psychological Issues of Human Computer Interaction in the Work Place, Amsterdam, 1987, pp. 257-274.

Dourish, P. Software Infrastructures. in Beaudouin-Lafon, M. ed. Computer Supported Co-operative Work, John Wiley & Sons, 1999, pp. 195-219.

Dourish, P. The Appropriation of Interactive Technologies: Some Lessons from Placeless Documents. Computer Supported Cooperative Work (CSCW) - The Journal of Collaborative Computing, 12 (4). 2003, pp. 465-490.

Dourish, P. and Edwards, W.K. A tale of two toolkits: Relating Infrastructure and Use in Flexible CSCW Toolkits. Computer-Supported Cooperative Work (CSCW), 9 (1), 2000, pp. 33-51.

Gruen, D., Rohall, S.L., Minassian, S., Kerr, B., Moody, P., Stachel, B., Wattenberg, M. and Wilcox, E., Lessons from the reMail prototypes. in Proceedings of the 2004 ACM conference on Computer supported cooperative work, (Chicago, Illinois, USA, 2004), ACM Press, pp. 152-161.

Hampton, C.A.: "Getting Started With Procmail", http://www.spambouncer.org/proctut.shtml, 8.1.2002.

Hanseth, O. and Lundberg, N. Designing Work Oriented Infrastructures. Computer Supported Cooperative Work: The Journal of Collaborative Computing, 10 (3-4), 2001. Pp. 347–372.

Henderson, A. and Kyng M.: "There's No Place Like Home. Continuing Design in Use", in: Design at Work, Lawrence Erlbaum Associates, Publishers, 1991, pp. 219-240.

Lieberman, H., Paternó, F. and Wulf, V. (eds.). End User Development. Kluwer, Dordrecht, NL, in press.

Mackay, W. E.: "Users and customizable Software: A Co-Adaptive Phenomenon", MIT, Boston (MA), PhD Thesis, 1990.

MacLean, A., Carter, K., Lövstrand, L., and Moran, T., "User-tailorable Systems: Pressing the Issue with Buttons", in: Proceedings of the Conference on Computer Human Interaction (CHI '90), April 1-5, Seattle (Washington), ACM-Press, New York, 1990, pp. 175-182.

Malone, T.W., Grant, K.R., Lai, K.-Y., Rao, R. and Rosenblitt, D., Semistructured Messages are Surprisingly Useful for Computer-Supported Coordination. in Proceedings of CSCW 88, (1988), Morgan-Kaufmann, pp. 311-334.

Malone, T.W., Lai, K.-Y. and Fry, C., Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. in Int. Conference on CSCW (CSCW'92), Toronto, Canada, 1992, ACM Press, pp.289-297

Morch, A., Stevens, G., Won, M., Klann, M., Dittrich, Y. and Wulf, V. Component-based technologies for end-user development. Communications of the ACM, 47 (9). 2004, pp. 59-62.

Mowshowitz, A: Virtual Organization. In: Communications of the ACM, 40-9, 1997, pp. 30-37

Nardi, B. A., "A Small Matter of Programming - Perspectives on end-user computing", MIT-Press, Cambridge et al., 1993.

Nielsen, J.: Evaluating the Thinking-Aloud Technique for Use by Computer Scientits. In: Hortson, H.R., Hix, D. (Hrsg.): Advances in Human Computer Interaction, Vol. 3, 1992, pp. 69-82. Nielsen, J.: Usability Engineering. AP Professional, New York, 1993.

Oppermann, R. and Simm, H., "Adaptability: User-Initiated Individualization", in: Oppermann, R. (ed.): Adaptive User Support – Ergonomic Design of Manually and Automatically Adaptable Software, Lawrence Erlbaum Ass., Hillsdale, New Jersey, 1994.

Paul, H., "Exploratives Agieren", Peter Lang, Frankfurt/M (Germany) 1994.

Picot, A., Reichwald, R., Wigant, R.: "Die grenzenlose Unternehmung - Information, Organisation und Management", 3. Aufl., Wiesbaden, Germany, Gabler, 1998.

Pipek, V., From Tailoring to Appropriation Support: Negotiating Groupware Usage Faculty of Science, Department of Information Processing Science (ACTA UNIVERSITATIS OULUENSIS A 430), University of Oulu, Oulu, Finland, 2005, 246 p.

Pipek, V. and Kahler, H. Supporting Collaborative Tailoring. in Lieberman, H., Paterno, F. and Wulf, V. eds. End-User Development, Kluwer, Dordrecht, NL, 2005, to be published.

Sandor, O., Bogdan, C. and Bowers, J., Aether: An Awareness Engine For CSCW. in 5th European Conf. on CSCW (ECSCW'97), Kluwer,, 1997, pp. 221-236.

Sutcliffe, A. and Mehandjiev, N. Special Issue on End-user development: tools that empower users to create their own software solutions. Communications of the ACM, 47 (9), 2004.

Rittenbruch, M., Kahler, H. and Cremers, A.B., Supporting cooperation in a virtual organization. in Proceedings of the international conference on Information systems, (Helsinki, Finland, 1998), Association for Information Systems, 1998, pp. 30-38.

Stiemerling, O., "Component-based tailorability", Ph.D.-Thesis, Institute for Computer Science III, University of Bonn (Germany), 2000.

Stiemerling, O. and Cremers, A.B. The EVOLVE Project: Component-Based Tailorability for CSCW Applications. AI & Society, 14. 2000, pp. 120-141.

Wulf, V.: "Let's see your Search-Tool! – On the Collaborative Use of Tailored Artifacts", in: Proceedings of Group '99, ACM Press, New York, 1999, pp. 50-60.

Wulf, V. and Golombek, B. Direct Activation: A Concept to Encourage Tailoring Activities. Behaviour & Information Technology, 20 (4), 2001, pp. 249-263.