

Supporting the change of cooperation patterns by integrated collaboration tools

Wolfgang Prinz¹, Nils Jeners¹, Rudolf Ruland¹, Matteo Villa²

¹ Fraunhofer FIT, Schloss Birlinghoven, 53754 Sankt Augustin, Germany

{Wolfgang.Prinz, Nils.Jeners, Rudolf.Ruland}@fit.fraunhofer.de

² TXT e-Solutions S.p.A., Via Frigia 27, 20100 Milano, Italy
Matteo.Villa@txt.it

Abstract. Although a number of new collaboration systems emerged of the last years, it is remarkable that email is still the most used collaboration application. However, this messaging based pattern of organizing collaboration causes a lot of problems like information and attachment overload and versioning problems. This paper discusses some of these problems as well as reasons why users are reluctant to switch to alternative cooperation means like document sharing in virtual project environments. Based on these observations we present tools developed with the EU funded Ecospace project that address these issue. These tools simplify the sharing process by combining and integrating sharing functionality with messaging, thus reducing the functional and cognitive distance between both environments.

Keywords: CSCW, Collaboration, Sharing, Architecture, Interoperability

1 Introduction and Motivation

Many cooperation support applications exist, but email is still the primary medium for electronic cooperation within and between organizations. Furthermore we can observe that email is often used for purposes that can be easier supported by other applications. However when we look for reasons to explain this misuse of email or the non-use of the more specialized systems instead of email, we can find a few simple reasons:

- **Simplicity and usability:** It is much easier to click the forward button to distribute an email or a document attachment than to start another application.
- **Loss of context:** If the user decides to start or use another application for his cooperation process the first step is to select the right process or group for his cooperation goal. This takes often several actions, i.e. finding and browsing to the right group or shared location, while typing the email address of a distribution list is much simpler.
- **Lack of interoperability:** Actually email is probably the most interoperable communication media. It works very reliably within and between organizations, while other systems like application sharing, shared workspaces and virtual project offices are almost not interoperable between different products.

- Lack of desktop presence: Email, together with a web-browser is the most present application on the user's desktop. Most other applications are started only on demand. Thus everything that does not appear in the email client is too far away from the user's attention and it thus not recognized.

The following table provides further examples of email usage and associated problems:

Table 1 – Email use and misuse

Cooperation process using email	Associated Problem	Specific application for this process
Exchange of a document as an email attachment between two people or through a distribution list	Multiple copies in all users email inboxes, versioning problem, information overload via distribution list	Shared folder of shared workspace systems.
Lengthly email discussions with multiple replies and threads	Email overload, loss of context	Discussion forums, Blogs
Status emails explaining the fulfillment of a task or the creation or modification of a document	Email overload, loss of context between task and email	RSS feeds, awareness tools
Availability emails, asking for the presence and availability of a user	Use of an asynchronous media to support a synchronous task, email overload	Presence systems

Beside an identification of problems the table also indicates cooperation applications and services that may be more suitable for a specific cooperation process. In the remaining of this paper we will present further approaches that have been developed in the context of the Ecospace¹ project to address these problems in the context of a cooperative activity [1].

2 From Messaging to Sharing

Information and communication overload is caused by the fact that information that should actually be shared is distributed by a communication media like email. Instead of storing relevant information in a dedicated location, users tend to forward and distribute that information via a distribution list. As a consequence this information is copied to all recipients. Often this leads to multiple replies in which various versions of a document are exchanged, resulting in a situation in which nobody actually knows about the location and version of the current document.

An important reason for this behavior is the fact that it is much simpler to forward a document by email than to store it in a shared repository such as a shared workspace or virtual project system. For example, storing a document that has been received as an attachment by email in a shared workspace requires the following steps: Saving the

¹ Ecospace is an Integrated Project (No. 035208), partly funded by the EC

attached document as a local file, starting the application or opening the appropriate web-site, navigation to the relevant project workspace, uploading the local file into the workspace. Compared to these steps the forwarding action is much simpler. This observation indicates that we need tools that support a paradigm shift from messaging to sharing. The following sections describe tools to simplify information sharing.

2.1 Smart Sharing Support

In recent years shared workspace systems [2] have become a widespread tool for the support of flexible and weakly structured cooperation in teams and communities. Typical examples for such systems are BSCW [3], Groove [4], or MS-Sharepoint [5]. Application areas for these systems are manifold such as the coordination of lectures, intra- and inter-organisational projects, or communities.

A shared workspace normally contains different types of information such as documents, pictures, URL collections, threaded discussions, or member profiles. The content of each workspace is represented as information objects arranged in a folder hierarchy. Since shared workspace systems do not impose a fixed structure on the workspace organization, each workspace can be organized according to the needs and requirements of the cooperating team. Most preferred structures for workspace organization are project structures (work packages, meetings) or organizational structures (departments, projects). Often structures that reflect both criteria are applied. However, the aim and intention of these structures is often not immediately visible to the users who share a workspace. Although workspace or folder descriptions can be used to describe the purpose of each workspace, users are often confused about the hierarchy, resulting in the effect that they have problems in finding the adequate folder to which they can upload a new document or where they can find the appropriate information. Although the users cooperate through a shared workspace, they often fail to develop a common understanding [6] or common conventions [7]. As a consequence they often complain about the complexity of a shared workspace and moreover they tend to turn back to email attachments for the sharing of documents, which contradicts the approach of a shared.

The shared workspace organizer [8] has been our first approach to address that problem. This approach makes use of the fact, that workspaces associate meta-information with each object (e.g. document owner, document mime type, creation date, version information, etc). The shared workspace organizer (SWO) uses that information in combination with a text analysis of all shared workspaces to propose suitable upload locations. After a user selects a local document the SWO suggests suitable upload locations based on a comparison of the document content with the indexed content of the shared workspace. Then the user could either select one of the suggested locations or he could use them for further navigation to the right place. In any case the SWO simplifies the upload of documents to a shared environment. The drawback of this solution is that it is mainly applicable to text documents and that it requires a continuous text mining and update of the shared workspace content. This consideration as well as experiments with a document context management tool [9] led to the development of a much simpler tool to support the smart upload of documents to a shared environment.

The smart upload tool is based on an extension of the standard document attributes by a context attribute that contains the original location of the document within the shared workspace. This context attribute is added by the shared workspace system and it contains information about the document location and the last modification date. This information is represented in XML as part of the new MS-Office document format that supports the extension of office documents by user defined attributes. The smart upload tool appears as an icon on the user's desktop. Whenever a user drags an object onto that icon, the upload tool analysis the document for the context information. If a document contains that meta-data it is interpreted and the document is stored at its original location in the shared workspace. If the tool detects that the document has been edited after the user downloaded the document from the shared workspace, it informs that user about a possible version conflict. The user can then decide to create a new version or to cancel the upload to solve the conflict manually.

This tool addresses primarily the *simplicity* problem identified above: it reduces the upload process that usually involves several navigation steps to a simple drag and drop action.

2.2 Sharing instead of forwarding attachments

The Sharing Support² tool further supports the paradigm shift from messaging to sharing. The tool itself is realized as an MS-Outlook plug-in that creates a new toolbar within the Outlook environment, which allows users to select one or more e-mail attachments for being saved in a shared environment and to provide additional context information. Instead of forwarding the email to a distribution list he can activate the "ECOSPACE Share" button. This results in pop-up window in which the user is able to choose the location of the shared folders and to add or modify meta-data (sender, receiver, subject, date, body, and comments) from the email.

Once metadata and location have been set, the plug-in will upload the documents to the corresponding BSCW shared repository using a set of pre-defined web-services [10]. The metadata is stored as attributes of the document, thus keeping the context of the document preserved.

This tool addresses two of the problems identified above: *simplicity* by integration of the sharing functionality into the messaging environment and *loss of context* by preserving the email context as metadata in the shared document attributes.

3 Cooperation Interoperability beyond Email

Shared Workspace platforms support a wide variety of collaboration functionalities such as document management, versioning, project blogging, shared todo lists or calendars. In many organizations SW platforms providing these functionalities are already implemented. However, cooperation processes occur not only within a single organization but more and more between various organizations that are involved in

² http://www.ami-communities.eu/wiki/ECOSPACE_Newsletter_No_4#Sharing_Support

common projects. In this kind of situation a strategy of using a single SW platform cannot be applied for the following reasons:

- Cooperating organizations have already different shared workspace technologies in use. Furthermore, it often happens that different departments or divisions within a single big organization use different shared workspace platforms.
- Access policies or licenses restrictions make it impossible to open the local shared workspace platform to external users.
- Partners are unable to agree on which SW platform to select.
- Users are reluctant to learn another SW platform while they have already one in use.

For users who are involved in several cross-organizational cooperation projects, this strategy of using one SW platform per project would imply that they have to learn as many different cooperation platforms as selected by the projects they are involved in. However, users are absolutely not motivated to invest much time in learning different tools that provide almost the same services, as they feel like wasting their time. This kind of particular situation is explaining why often users turn back to the most simple cooperation tool they know the best such as email attachment to "share" a document (though, in this case it is more about "sending a document" than really "sharing a document"). For sure, as the emailing protocol is a standard it gives the freedom to organizations and users to select the email server and client they prefer which is a very good illustration of the power of standardization.

Consequently, it would be really great if SW platforms could have a good level of interoperability enabling users to access posted objects on other SW platforms from their own SW client whatever is the SW server.

Interoperability between SW platforms is addressed by the ECOSPACE Collaborative Working Environments (CWE) reference architecture. The most important components in this CWE reference architecture are the identification of basic services for each collaboration service and the definition of an exchange/communication protocol enabling the representation and interpretation of the workspace object meta-data from different SW platforms. Each SW application implements standardized basic services enabling:

- Access (read/write) to workspace objects,
- Retrieval of the workspace organization (i.e. the folder structure) and user information;
- Exchange of objects' meta-data through the use of the SIOC (Semantically-Interlinked Online Communities) format [11].

According to this interoperability approach, each SW service is now offering the same web services to access and modify objects' meta-data. Access to this information is provided by new interface components in each shared workspace user interface. These components provide access to the workspaces of remote systems in a transparent way and in the same look and feel as if they were local workspaces while they are external.

The following use case is an illustration of interoperability among 3 different Shared Workspace platforms (BSCW, Business Collaborator and SharePoint). Three companies are starting a joint project and they wish to use their own available SW platform (BSCW, Business Collaborator or SharePoint) for both their own developed

objects and joint developed objects. Company A uses BSCW, company B uses Business Collaborator and Company C uses SharePoint. In a first step, each company creates a project workspace in their own SW platform. Afterwards, they reciprocally invite participants from the cooperating organizations as external users to their own project workspace. Finally, they exchange the web service address (the access path to their local workspace). The way this access path is used in the respective SW platform depends on the local implementation. The implementation for BSCW is described below.

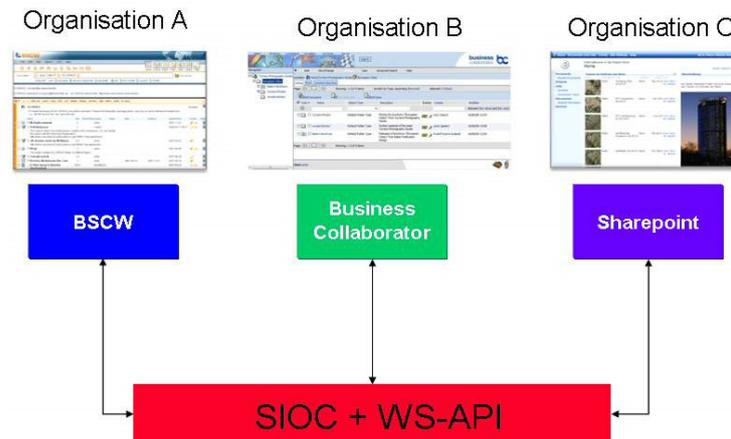


Fig. 1: Shared Workspace Interoperability based on the SIOC format and Web-Services API

A user in BSCW creates a new folder which is a "shadow" folder type by providing the access path to the external folder, located into another SW platform, as well as the authentication information to access the remote SW platform. This shadow folder is created as a "sub-folder" of the local project workspace which is the "top" folder. When a user opens this shadow folder then a special background color indicates that the information provided in this folder is not stored into the local SW platform but accessed from a remote SW server. Furthermore, the shadow folder provides the same look and feel that users are used to have on their own SW platform. It means that users can access the external objects information that are actually stored into another SW platform in a very transparent way without to have to learn other SW platforms.

Obviously the functionalities offered by this shadow folder type are limited to the necessary set of services required to access and modify objects. In this implementation example, the current limitation is that other advanced functionalities provided by some SW platforms (i.e. rating, annotating, tagging) cannot be used. Therefore, future work should address this issue by the development of a more advanced protocol that enables also the exchange of service capabilities between the different SW platforms.

This concept addresses the *interoperability* problem identified above: it provides a references architecture in combination with the definition of web-services and protocols to enable transparent access between different shared workspace systems.

4 Modularization of Cooperation Services

Current cooperation solutions are very often still monolithic entities. Although they often provide a user interface that can be adapted to the user's needs, they cannot be completely reconfigured or re-assembled by the user. This is in contrast to the self-service and customization mentality of the users who are used to configure their working environment by downloading and installing the appropriate tools.

Applying a reference architecture [12], including the definition of basic cooperation services, we can open the door towards an individually configurable cooperation solution. This is achieved by realizing a set of interaction widgets to support basic collaboration services. It means that users can configure their cooperation environment by the selection of those collaboration services that are most appropriate for their specific tasks. This motivation has led to the development of a set of CWE widgets that correspond to basic collaboration functions.



Fig. 2: Comparison of a monolithic user interface and a widget based collaboration portal.

Figure 2 is presenting, in the left part, a screenshot of BSCW user interface and in the right part an example of a widget based application which is a portal page in iGoogle that includes different widgets for basic collaborative services. Using that portal page to combine different cooperation widgets any project participant can personalize his private activity space and information visualization to support his collaborative activities in accessing different services of the Shared Workspace platform.

This widget approach for a cooperation environment addresses the *desktop presence* problem identified above: it enables users to add different cooperation functionalities as widgets to their desktop, thus staying aware of the ongoing activities in the related cooperation processes.

5 Summary and Conclusion

This paper presents different approaches to overcome current cooperation problems that originate in the use and misuse of email as the primary communication media. At the beginning we have identified four main problems and for each problem a possible solution has been presented:

Simplicity and usability as well as **loss of context** is addressed by simple sharing tools that reduce the overhead of navigation within a shared workspace to simple drag and drop operations. **Lack of interoperability** is addressed by a shared workspace interoperability approach based on a reference architecture including shared workspaces and a SIOC based exchange protocol. **Lack of desktop presence** can be overcome by the approach to widgetize a complex cooperation environment to enable users to configure their own environment as their desktop.

We believe that these concepts can provide a first step towards a change of cooperation patterns from messaging to sharing, thus reducing communication overload and complexity.

References

1. Harrison, B.L., A. Cozzi, and T.P. Moran, Roles and relationships for unified activity management in Proceedings of the 2005 international ACM GROUP conference. 2005 ACM Press Press: Sanibel Island, Florida, USA p. 236-245
2. Hans Schaffers, T.B., Marc Pallot, Wolfgang Prinz, ed. The Future Workplace - Perspectives on Mobile and Collaborative Working. 2006, Telematica Instituut, The Netherlands. 112.
3. Appelt, W. WWW Based Collaboration with the BSCW System. in SOFSEM'99. 1999. Milovy, Czech Republic: Springer Lecture Notes in Computer Science 1725.
4. Groove-Networks, Groove. 2005.
5. Microsoft, Microsoft Sharepoint. 2005.
6. Bannon, L. and S. Bødker. Constructing Common Information Spaces. in ECSCW'97: Fifth European Conference on Computer Supported Cooperative Work. 1997. Lancaster, UK: Kluwer Academic Publishers.
7. Prinz, W., G. Mark, and U. Pankoke-Babatz. Designing Groupware for Congruency in Use. in CSCW '98: ACM Conference on Computer Supported Cooperative Work. 1998. Seattle: ACM Press.
8. Prinz, W. and B. Zaman. Proactive Support for the Organization of Shared Workspaces Using Activity Patterns and Content Analysis. in GROUP'05: 2005 International ACM SIGGROUP Conference on Supporting Group Work. 2005. Sanibel Island, Florida, USA: ACM Press.
9. Vonrueden, M. and W. Prinz. Distributed Document Contexts in Cooperation Systems. in CONTEXT 2007, LNAI vol. 4635. 2007: Springer-Verlag Berlin Heidelberg
10. Prinz, W., et al. ECOSPACE – Towards an Integrated Collaboration Space for eProfessionals. in CollaborateCom 2006. Atlanta: IEEE press.
11. Breslin, J. and S. Decker, The Future of Social Networks on the Internet: The Need for Semantics. IEEE Internet Computing, 2007. 11(6): p. 86-90.
12. Peristeras, V., et al., Towards a Reference Architecture for Collaborative Work Environments. International Journal of e-Collaboration, 2009: p. to appear.