Rendezvous

Federico Naso, Alejandro Fernández
LIFIA, Laboratorio de Investigaciones y Formación en Informática Avanzada
UNLP, Universidad Nacional de La Plata, Argentina
[Federico.Naso, Alejandro.Fernandez]@lifia.info.unlp.edu.ar

Abstract

Asynchronous groupware supplies no presence indicators, and does not commonly provide means to start synchronous collaboration. In cases where users are on-line simultaneously, the mentioned characteristics result in an inefficient use of time and effort. The solution documented in this pattern extends the asynchronous groupware with a mechanism to inform users of on-line presence in other available, synchronous groupware. In this way users are aware of the event when his/her peers are able to collaborate synchronously by means of other (synchronous) tools

Intent

To inform a user in an asynchronous groupware system of the event when his/her peers are able to collaborate synchronously by means of other tools (that is, Synchronous groupware).

Also know as

Meeting place, Meeting, Confluence.

Problem

Asynchronous groupware supplies no presence indicators, and does not commonly provide means to start synchronous collaboration. In cases where users are on-line simultaneously, the mentioned characteristics result in an inefficient use of time and effort.

Example

Imagine an open-source software development team at work. Programmers, testers, designers, and the project leader are geographically distributed. They constantly communicate via e-mail. Two hours before the workday is over, the development team finds a problem that prevents team members to continue. The programmers send a description of the problem via e-mail to the design leader. The design leader is the one with the final word on the solution. Aware of the asynchronous nature of e-mail, the developers postpone all pending activities to the next work day. At the time of the problem, the design leader is working in his workplace. However, he does not pay attention to incoming e-mail. In such a scenario, a problem which could have been solved immediately is postponed to the next workday, thus resulting in the loss of valuable work hours.

Forces

Asynchronous collaboration is valuable because its give participants time to elaborate their contributions. Moreover, it gives them the chance of contributing when they considers it most convenient.

In some cases, synchronous interaction reduces the effort and time required to solve a shared task.

Synchronous collaboration requires what participants are simultaneously on-line, either by chance or intentionally.

The integration of collaboration services takes the best thing from each one. This way empowers application where they are integrated.

Synchronous collaboration services commonly have an indicator of on-line presence. In contrast, asynchronous collaboration services lack one. Presence indicators provide information about user availability (e.g. free, occupied, off-line, etc).

Solution

Extend the asynchronous groupware with a mechanism to inform on-line presence from the available synchronous groupware.

Participants

Asynchronous groupware service: Allows asynchronous collaboration. Examples of asynchronous groupware are E-mail, forums, and web-logs. Asynchronous groupware commonly has a mechanism to identify users. In the previous examples these mechanisms are the e-mail address or the user names.

Synchronous groupware service: Allows synchronous collaboration. Examples of asynchronous groupware are chat, instant messengers, and white boards. Synchronous groupware commonly has a mechanism to identify users. In the previous examples these mechanisms are the nickname or the user names.

Presence service: maintains and supplies information about the on-line presence (also known as the on-line status) of users of a synchronous groupware service (E.g. on-line; off-line; busy; free for chat).

Collaborations

The user logs in to the synchronous groupware service, eventually changes state, and finally logs off. The synchronous groupware service communicates changes of the on-line presence of users to the presence service. Additionally, the synchronous groupware service queries the presence service about the on-line status of other users. The resulting information is provided to the user.

The user starts the asynchronous groupware service. Each time a contribution made by someone else is to be shown to the user, the asynchronous groupware service queries the presence service for the on-line status of the contributor. This information is displayed together with the contribution.

The user observes that the author of an asynchronous contribution is available for synchronously interaction (E.g., free for chat).Consequently, the user starts synchronous collaboration.

Implementation details

There are cases where the mechanism for the identification of users used by the synchronous groupware service differs from the one used by the asynchronous groupware service. For these cases, a mapping of ids needs to be performed. Depending on the involved groupware services, this mapping may be done by one of them, or by the presence service.

This pattern is easier to implement when: 1) the services to integrate are offered by a same system; 2) the services to integrate are offered by independent systems that share a common communication protocol, and work on connected networks.

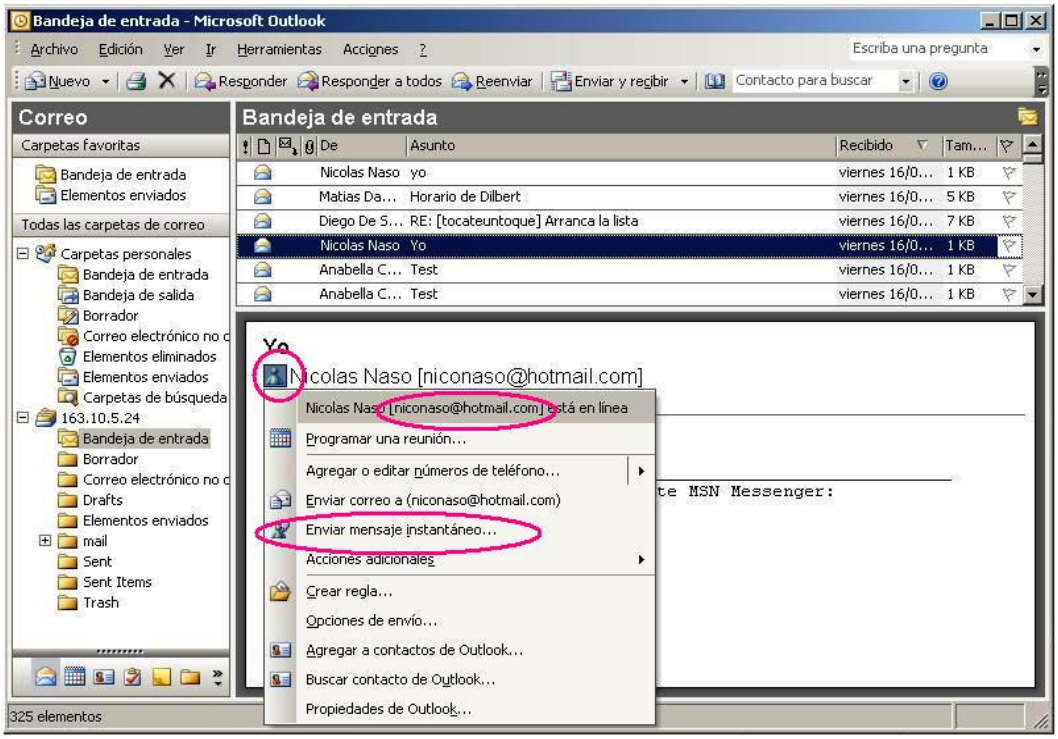

Fig. 1

Applicability

Use rendezvous pattern when:

- Distributed users supported by groupware perform a task that mainly involves asynchronous work. However, the task eventually requires that affected users collaborate synchronously for short periods of time (e.g., to agree on the interpretation of a requirement). You want to show when/if it is possible to contact them synchronously.

- Distributed users work on artifacts (e.g., documents, pieces of source code) that are the result of previous work. These artifacts hold relations to the team members that contributed (e.g., authors of the document). As a part of what is being done with the artifacts, it becomes necessary to shortly and immediately contact the original contributors (e.g., for clarification). You want to show when/if it is possible to contact them synchronously to speed up the process.

Consequences

This pattern offers the following benefits:

- Reduce time and effort used in synchronizing users who wish to work of synchronous way.

- Offers a vision of the services that must be integrated to offer support of interaction between asynchronous and synchronous applications.

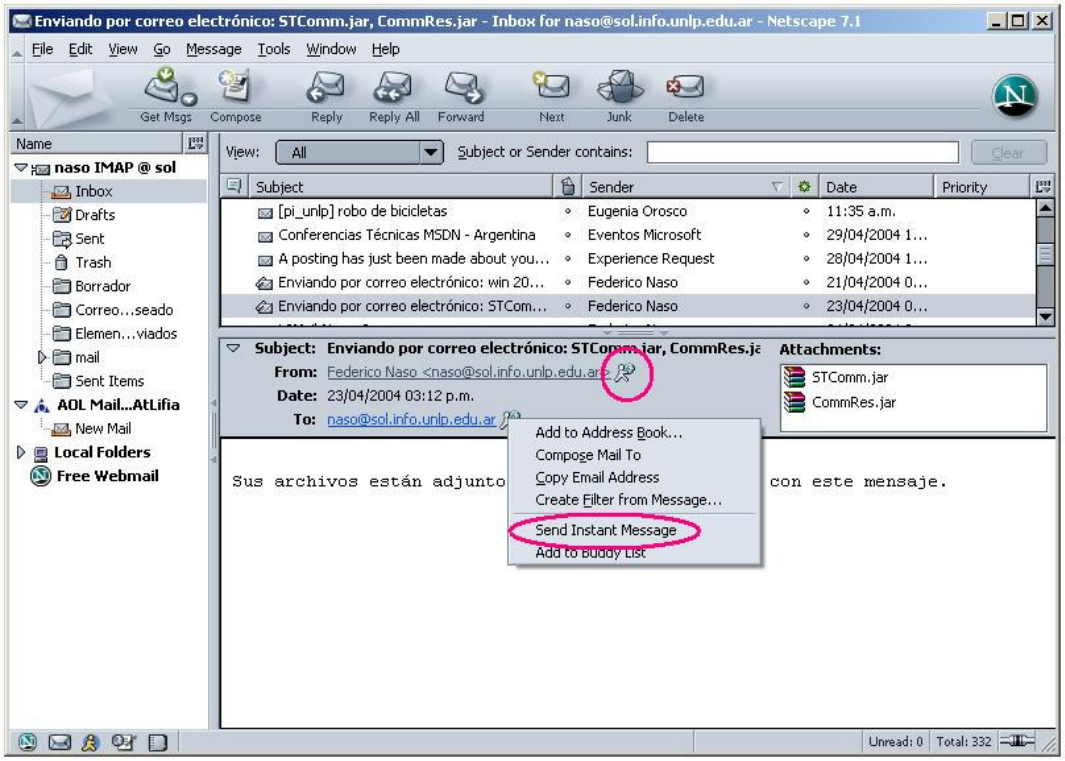

Fig. 2

Known uses

The Microsoft Outlook™ e-mail client integrates with ICQ. The presence service is provided by the ICQ network. The mapping between the ids used by outlook (i.e., e-mail addresses) and the ids used by ICQ (i.e. UIN) is done via the ICQ contact information. A single ICQ UIN can map to several e-mail addresses. When an e-mail is selected for display in outlook, the ICQ contact list is searched for a mapping. If a mapping is found the on-line status of the given UIN is displayed.

Outlook similarly integrates with Microsoft's MSN© Messenger (see Fig 1). The presence service is provided by the MSN network. The MSN messenger uses e-mail addresses from the MSN network (e.g. hotmail.com, msn.com and passport.com) for users' ids, therefore the mapping is straight forward for e-mails send by users of the msn network. No integration is available for e-mail users of other providers.

Netscape 7.1 integration with AOL® Instant Messenger™ (AIM®) (see Fig. 2) work as in the case of to Microsoft Outlook™ with ICQ. However, the mapping (only one e-mail per AIM screen name) is done via the Netscape address book. The presence service is provided by the AOL AIM network.