

PRIMInality: Towards Human-Centred Instant Messaging Infrastructures

Tom Gross, Christoph Oemig

Faculty of Media, Bauhaus-University Weimar, Germany

Abstract

Instant messaging systems facilitate the coordination of geographically distributed users through support for presence and availability information, and ad-hoc conversations among online users. However, despite their widespread use they seem to be designed and developed from a technology-push, rather than a user-centred design perspective. In fact, concerning the subtleties of human behaviour and human perception of others there is a broad gap between the body of knowledge in sociology, and the functionality of these systems. In this paper we present concepts for the support of advanced and nuanced online states; for the automatic and precise inference of the online states based on data from software sensors and hardware sensors; and the adaptation of the system behaviour to the inferred online state. Furthermore, we describe the implementation of these concepts on top of two open platforms for the rapid and easy development of sensor-based, and instant messaging infrastructures.

1 Introduction

Instant messaging (IM) systems are wide-spread systems providing presence and availability information in order to facilitate coordination, and ad-hoc conversations among online users (Erickson et al. 2004; Fussell et al. 2004; Gross & Prinz 2004; Nardi et al. 2000). Basically, they allow users to register, log in and out, express their own presence and availability via an online status as well as see the online status of other users and send instant messages or start synchronous text chats with other users who are online at the same time. Many IM systems are available: single protocol clients such as ICQ, AOL Instant Messenger, Yahoo! Messenger, or MSN Messenger typically only support their native protocol for the communication between clients and server; and multi-protocol clients such as Gaim, Trillian, Miranda, Adium, Fire, or Proteus typically support several protocols in parallel (Wikipedia 2005).

Despite their widespread use IM systems seem to be designed and developed from a technology-push, rather than from a *user-centred design* perspective. In fact, concerning the subtleties of human behaviour and human perception of others there is a broad gap between the detailed body of knowledge in sociology (e.g., Goffman 1959) as well as human communication (Short et al. 1976; Watzlawick et al. 1967) and the functionality of IM systems.

Here we want to focus on two particular gaps: online states and personalities. Most existing IM systems have quite mechanic and simple mechanisms for dealing with online states. They have a very limited number of online states (typical online states are Available, Away, Not Available, Occupied, Do Not Disturb, or Offline), not covering the myriad of actual state of human presence and availability; and either the user has to set the state manually, or the system has to adapt the states automatically based on simple assumption that sometimes can be misleading (e.g., in most IM systems a user who has been inactive with keyboard or mouse is automatically put from Online to Away; in the case where a user is at her desk without using the computer, this assumption is incorrect). Additionally, many IM systems do not provide adequate support for the selective disclosure (Palen & Dourish 2003) of personal information to other users (e.g., most single-protocol IM systems only allow one general setting for entering and sharing personal data that is subsequently made available to all other users in the same way).

In order to build IM systems from a user-centred design perspective, system developers need adequate platforms to build systems and infrastructures, and flexible mechanisms for adapting and extending them. Currently, most IM software consists of clients, which are available for download and local installation, and of servers, which are usually run and administrated by commercial companies. Some examples of IM servers that are available for local installation are various Jabber servers. Extendibility is important for clients and servers. Yet, in general, IM clients are available as binary versions, which can hardly be extended. Some exceptions are: Gaim (SourceForge.Net 2005a), Miranda (SourceForge.Net 2005b), and Trillian (Cerulean Studios 2005), which can be extended via plug-ins; Gaim and Miranda are Open-Source. And, IM servers are often not available at all. Some positive examples of servers available in source code are Jabber servers (cf. Niedermann 2005 for an overview).

We have developed the PRIMInality infrastructure, which provides the following solutions addressing the above-mentioned problems: *A sophisticated IM infrastructure*: PRIMInality provides an infrastructure that allows modelling and implementing social nuances that are part of our everyday life, yet not supported by existing infrastructures. *Open platforms*: PRIMInality is realised on top of the open PRIMI (Platform for Research on IM Infrastructures) platform, and Sens-ation platforms. *Rapid and easy development*: PRIMInality has an elegant plug-in mechanism for easy development of clients and extensions concerning new IM protocols, new operating systems and platforms, and new user interaction; easy development of servers and extension of servers concerning new IM protocols, and calculations and propagations of online states; as well as application programming interfaces for the communication with other applications and infrastructures. In the remainder of this paper we present the human-centred design concepts in PRIMInality. Additionally, we provide the technical background describing the infrastructure's architecture.

2 Concept

In this section we briefly motivate the requirements of adequate support for social nuances in human-centred IM infrastructures, and present the design of PRIMInality for meeting these requirements.

2.1 Requirements

Goffman (1959) did many detailed analyses of the subtleties of human behaviour and human perception of others in social interaction. He saw social interaction mainly as a performance, which is shaped by the audience and the environment. At the same time humans analyse each other in order to infer their character, role, status, mood, and so forth. Since humans are aware that they are monitored and analysed by others, they try to deliberately construct their social identity by presenting others with a front—they manage their identity, and try to control the recipients' impressions. Furthermore, Goffman pointed out that communicative processes are generally asymmetrical, because the audience can perceive and analyse more details of the actor than the actor is able to control (Krallmann & Ziemann 2001).

Therefore, if IM infrastructures want to support real-life presence and availability, as well as ad-hoc social interaction, they require taking this human behaviour into account. That is, they need to allow users to manage their own online identity, like they would manage their social identity in real-life (cf. Goffman 1959; Short et al. 1976; Watzlawick et al. 1967).

For this purpose, it is important that the presence and availability information is accurate—resulting in the need for a variety of differentiated online states. Furthermore, these online states need to reflect the user's real situation. The setting of online status should cause users minimal effort—therefore, IM infrastructures need to automatically capture both information about their users in the electronic world by software sensors, and in the real world by hardware sensors. Additionally, users should be able to manage their identities in dependence of the social group they are interacting with. That is, they should be able to disclose certain personal information to some social groups, but keep the same information private to other social groups.

Subsequently we describe how the concepts of PRIMInality support these requirements.

2.2 Extended Online States

The PRIMInality infrastructure supports a whole range of online states; it is able to precisely infer online states based on data from software sensors and hardware sensors; and it adapts its behaviour to the detected online state.

The online states in PRIMInality are inferred from the data of four different sensors. Two software sensors in the PRIMInality client check whether the user is currently logged in and whether the user is involved in an online chat. Two hardware sensors capture movement at

the user's computer desk, and in the other parts of the user's office. In order to simplify we define movement as a sensor value of three or higher on a scale from zero to ten. Additionally, the time interval is set to two minutes—that is, only if the user has left for more than two minutes, the movement value is set to zero.

Using binary values for four sensors yields 4^2 (i.e., 16) possible states (cf. Table 1). Four of them turned out to be impossible, since a user cannot have open chat windows if she is not logged in (cf. rows 5-8 in Table 1). The columns indicate the binary results of the calculated values for the four sensors, the name and descriptions of the online states, and the name of the user interface adaptation profile of the PRIMInality client.

Online states for users who are not logged into the PRIMInality client are: *Away*: similar to the Away status of existing IM systems, the user is not active in the PRIMInality client application; however, beyond existing IM systems the two movement sensors tell us that the user is not in her office (in existing IM applications the user might be on an Away status while she is not active at the computer, yet present in the office). *Moving in Office*: the user is in her office, but not using PRIMInality. *At Computer*: the user is at the computer desk, but not using PRIMInality. *In Office / with Company*: since there is movement at the computer desk and in other areas of the office, we assume that the user is at the computer desk and has company.

And, online states for users who are logged into the PRIMInality client are: *Idle*: the user is logged in, but has left the office. *Idle / Moving in Office*: the user is logged in, but is moving in her office (e.g., standing at a book shelf). *Available*: similar to the Available status of existing IM systems, this means that the user is logged in to the PRIMInality client; additionally the user is sitting at her computer desk and is really available. *Do Not Disturb (DND)*: similar to the DND status of existing IM systems, this means that the user is using the PRIMInality client, but does not want to be disturbed; this status is set automatically when the user is logged in and at her desk, but has company (based on the movement in the other areas of the office). *Shortly Gone*: the user has started the PRIMInality client and its chat feature, but there is no movement in the room at all—here we assume that the user has just left for a few minutes, but will return soon (e.g., she is quickly getting a coffee). *Side Chat*: the user is using the PRIMInality client and its chat feature, but is moving through the office; here we assume that the user is still interested in the chat, but has some other tasks (e.g., went to the bookshelf, while waiting for a reply of the remote party). *Focused Chat*: the user is using the PRIMInality client and its chat feature, and is sitting at her computer desk. *Focused Chat / with Company*: the user is using the PRIMInality client and its chat feature, and is sitting at her computer desk; at the same time the user has some company (e.g., the user is in a discussion with a visitor and quickly asks a third person for some information via chat).

This state information is used for adapting the user interface of the PRIMInality client. The five adaptation profiles are: *Normal*: standard sized windows, average volume of chat alerts; *Mini*: shows just necessary fields for a login, in the background; *Loud*: normally sized windows, louder alerts for incoming messages; *Silent*: normally sized windows, envelope icons to silently alert the user of incoming messages, sounds turned off; and *Auto respond*: automatic answers to incoming messages stating the user is idle.

	Sensors				Online states		GUI
	PRIMI- nality: logged in	PRIMI- nality: open chat window	Com- puter desk: move- ment	Other office area: move- ment	Online status: name	Online status: description	PRIMI- nality: Adap- tation Profile
1	0	0	0	0	<i>Away</i>	Not in office	Mini
2	0	0	0	1	Moving in Office	In office, not in PRIMI-nality	Mini
3	0	0	1	0	At Com- puter	In office, at com- puter	Mini
4	0	0	1	1	In Office/ with Company	In office, at the computer, with company	Mini
5	0	1	0	0	-	-	-
6	0	1	0	1	-	-	-
7	0	1	1	0	-	-	-
8	0	1	1	1	-	-	-
9	1	0	0	0	<i>Idle</i>	Available, not in office	Auto respond
10	1	0	0	1	<i>Idle/ Mov- ing in Office</i>	Available, in of- fice, not at the computer	Loud
11	1	0	1	0	<i>Available</i>	Available, in of- fice, at the com- puter	Normal
12	1	0	1	1	<i>Do Not Disturb (DND)</i>	Online, in Office, with Visitor	Silent
13	1	1	0	0	Shortly Gone	Has been chatting, gone for a moment	Loud
14	1	1	0	1	Side Chat	Chatting not at the computer	Loud
15	1	1	1	0	Focused Chat	Chatting	Normal
16	1	1	1	1	Focused Chat/ with Company	Chatting at the computer, with company	Silent

Table 1. Overview of sensor values, corresponding online states, and adaptation profiles (online states with the same name as in existing IM systems are set in italics).

2.3 Personalities

While the previous section describes the case of a single online identity, PRIMInality allows a user to create several online identities. Online identities are managed by personalities. A

personality is basically a collection of personal data that a user wishes to share with others, and a list of online buddies who have access to this data. This way it is possible that a user selectively disseminates specific information to individual social groups (e.g., shares private data with private friends, but does not reveal the same data to working colleagues).

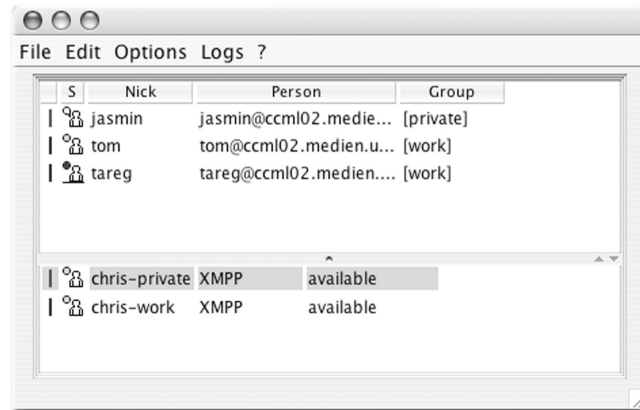


Figure 1. Screenshot of the PRIMInality client

Figure 1 depicts a screenshot of the PRIMInality client we are currently using. It shows the two personalities—chris-private, and chris-work—of the current user chris, and three buddies. The details about the buddies are: jasmin is a private buddy (light-grey line on the left) and is currently in the Focused Chat status (white speech bubble); tom is a work buddy (dark-grey line on the left) and is currently in the Available status (very light-grey circle); also tareg is a work buddy (dark-grey line on the left), who is currently in the Do Not Disturb (DND) status (very dark-grey circle).

For the combination of online states and personalities, the user has a table such as in Table 1 for each personality. In order to selectively disseminate information the user may decide which of the online states (i.e., rows) should be propagated to other users (i.e., for each personality a user can specify which row is active). Online states that are not selected are not propagated; instead, an online state Unknown is presented to the other users. However, for the user herself the online state information is still used for adapting the user interface.

3 Implementation

The above concepts were implemented by means of our two platforms: PRIMI (Platform for Research on Instant Messenger Infrastructures) and Sens-ation. In this section we first describe the concepts of the open platforms, and the rapid and easy development of infrastruc-

tures with these platforms. We then provide a brief overview of the overall architecture. The major part of the infrastructure is implemented using the PRIMI platform; the Sens-ation platform is mainly used for the hardware sensors and for the storage of sensor data and online states. The PRIMI platform covers three core issues: data handling; communication protocols; and application integration. These aspects are considered on the client and server side. The platform is used to build clients and servers that support existing and custom IM protocols. It is open, extensible and thus very flexible. It can be integrated into other applications (e.g., as IM support module) or can integrate other applications. In order to achieve rapid development and deployment the PRIMI platform provides two major concepts: first, separation of concerns: a plugin mechanism allows for easy extensions (GUIs and protocols are components separating concerns in application development); second, standards: serve as a common language among domain developers, and facilitate naming and requirements.

Figure 2 shows the overall architecture consisting of two infrastructures: the PRIMIBase infrastructure was developed with the PRIMI platform, and the SensBase infrastructure was developed with the Sens-ation platform. The left part shows the SensBase components of the architecture, and the right side shows the PRIMIBase components. The top depicts the client side, which consists of the software and hardware installations at the users' sites. The lower portion of Figure 2 shows the server side.

PRIMIBase is a plugin-based infrastructure written in the Java programming language for IM. User interfaces and communication protocols are deployed as plugins as part of a central platform that offers all necessary services, especially for plugin handling and logging. The implementation incorporates the Proxy pattern (Gamma et al. 1994) (i.e., plugin components are wrapped into proxy objects, which carry out the actual logging in all of the interface's methods). Currently PRIMIBase utilises two plugins, a user interface (a PRIMInality client) and a communication protocol (Extensible Messaging and Presence Protocol (XMPP); (Saint-Andre 2004a) and RFC 3921 (Saint-Andre 2004b)).

SensBase provides a generic infrastructure mostly written in the Java programming language that facilitates the registration and management of hardware sensors, the capture and storage of sensor values via these sensors, and the retrieval of information on sensors and present and past sensor values via various gateways (e.g., Web-Service, XML-RPC, Sockets, CGI).

The connection of PRIMIBase and SensBase works as follows: in our three offices (e.g., Location A, Location B, Location C) we installed sensors capturing the movement at the computer desk, and in the rest of the room (we use ESB sensor boards (ESB 2004)). These two sensors send their sensor data to the SensBase server. When a user starts the PRIMInality client, PRIMIBase registers with the SensBase server, and permanently queries for movement data in the office (the mapping of the location of the sensors and the client is currently done by means of a simple table where the IP-numbers of our computers are stored together with their physical locations). From the sensor values the PRIMInality client calculates the user's online status according to Table 1 and publishes this status back to the Sens-ation server to be seen by other users. This is done for every personality of the user with its individual table. Thus, other users can query the online states of their buddies from the SensBase server. Finally, the PRIMInality client applies the adaptation profile corresponding to the user's online status (we use the State pattern (Gamma et al. 1994) to change the appli-

cation's behaviour at runtime; all of the above five adaptation states are derived from a common state interface). The chatting between PRIMInality clients works via the XMPP server. On a whole, in Figure 2 the white rectangles are optional alternate plugins for user interface and communication.

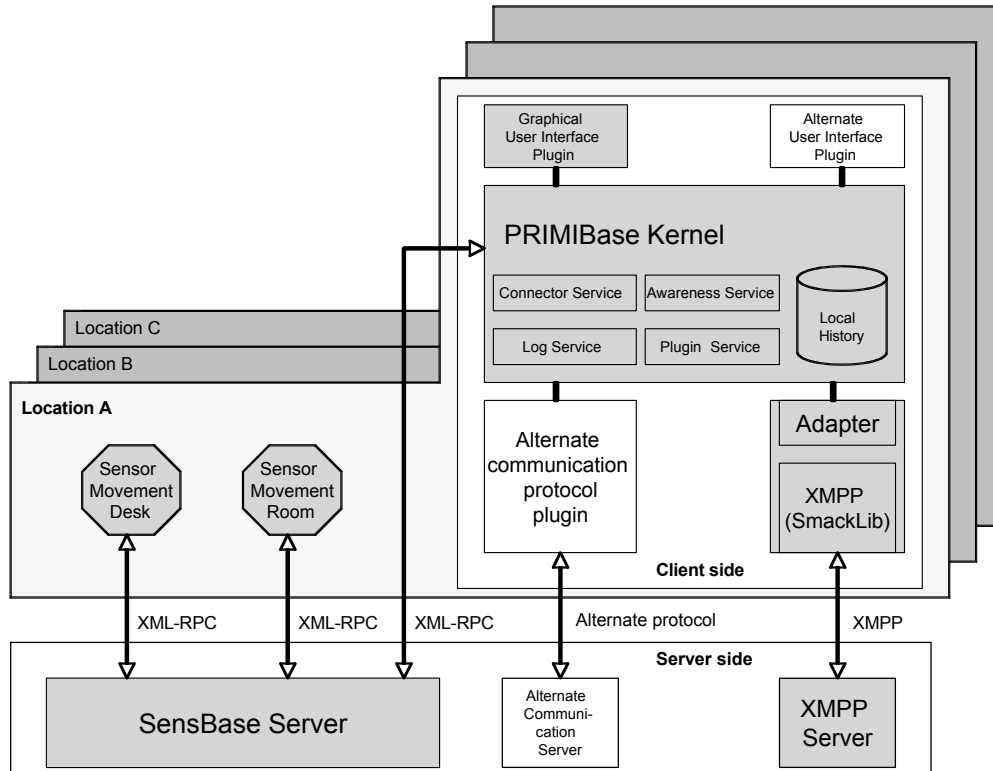


Figure 2. Overall infrastructure architecture.

4 Conclusions

In this paper we have motivated the need for human-centred IM systems as means to support the coordination among geographically distributed members of workgroups. We have described the concepts, which are based on requirements for a user's identity management and selective disclosure rooted in sociology. The concepts include novel approaches for defining, inferring, and reacting to extended online states, and for protecting privacy through person-

alities. And we have sketched the implementation based on two open platforms supporting the rapid and easy development of sensor-based infrastructures, and IM infrastructures.

Implementing these concepts we faced several challenges that only emerge when dealing with multiple personalities: Here we can only briefly describe two examples. First, the adaptation of the user interface of a user who is logged in with several different personalities at the same time is difficult, since there is only one user interface for all personalities. As a solution the user interface is adapted to the most recently active personality. Secondly, users should be able to control both software and hardware sensors. In PRIMInality the users can implicitly switch software sensors on by starting the PRIMInality client, and switch them off by quitting the client. The can manually switch hardware sensors on and off, which proved sufficient for the current version.

The concepts described in this paper clearly have their limitations. For instance, currently we need to install two ESB sensor boards in each office. For other offices the users can manually enter their online status, but this is cumbersome. Furthermore, our approach is simplified in order to reduce complexity (e.g., concerning the thresholds for the sensor data). Finally, we use a fixed set of sensors; further infrastructures with other sensor types and more sensors have yet to follow.

On a whole we hope that to some extent our approach goes beyond the social phenomena described by Goffman (1959), since—as opposed to the asymmetry in real-life communicative processes described by Goffman—our approach empowers the users to completely control their online appearance and thereby enforces symmetry among all online users.

References

- Cerulean Studios. Trillian and Trillian Pro Instant Messengers. <http://www.ceruleanstudios.com/>, 2005. (Accessed 22/2/2005).
- Erickson, T., Huang, W., Danis, C. and Kellogg, W.A. A Social Proxy for Distributed Tasks: Design and Evaluation of a Working Prototype. In Proceedings of the Conference on Human Factors in Computing Systems - CHI 2004 (Apr. 24-29, Vienna, Austria). ACM, 2004. pp. 559-566.
- ESB. Sensorboards Documentation. Freie Universitaet Berlin, Germany, http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb_net/ESB/sensorboards/doc/html/index.html, 2004. (Accessed 24/2/2005).
- Fussell, S., Kiesler, S., Seetlock, L.D., Scupelli, P. and Weisbrand, S. Effects of Instant Messaging on the Management of Multiple Project Trajectories. In Proceedings of the Conference on Human Factors in Computing Systems - CHI 2004 (Apr. 24-29, Vienna, Austria). ACM, 2004. pp. 191-198.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1994.
- Goffman, E. The Presentation of Self in Everyday Life. Doubleday Anchor Books, N.Y., 1959.
- Gross, T. and Prinz, W. Modelling Shared Contexts in Cooperative Environments: Concept, Implementation, and Evaluation. Computer Supported Cooperative Work: The Journal of Collaborative Computing 13, 3-4 (Aug. 2004). pp. 283-303.

- Krallmann, D. and Ziemann, A. Grundkurs Kommunikationswissenschaft (Introduction of the Science of Communication; in German). Fink, Munich, 2001.
- Nardi, B.A., Whittaker, S. and Bradner, E. Interaction and Outeraction: Instant Messaging in Action. In Proceedings of the Conference on Computer-Supported Cooperative Work - CSCW 2000 (Dec. 2-6, Philadelphia, PE). ACM, N.Y., 2000. pp. 79-88.
- Niedermann, F. Open-Source Jabber Server Comparison Chart. <http://www.jabber.org/admin/jsc/>, 2005. (Accessed 22/2/2005).
- Palen, L. and Dourish, P. Unpacking 'Privacy' for a Networked World. In Proceedings of the Conference on Human Factors in Computing Systems - CHI 2003 (Apr. 5-10, Minneapolis, Minnesota). ACM, 2003. pp. 129-136.
- Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Core (RFC 3920). <http://www.ietf.org/rfc/rfc3920.txt>, 2004a. (Accessed 20/2/2005).
- Saint-Andre, P. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence (RFC 3921). <http://www.ietf.org/rfc/rfc3921.txt>, 2004b. (Accessed 20/2/2005).
- Short, J., Williams, E. and Christie, B. The Social Psychology of Telecommunications. Wiley, London, UK, 1976.
- SourceForge.Net. Gaim - A Multi-Protocol Instant Messaging (IM) Client. <http://gaim.sourceforge.net/>, 2005a. (Accessed 22/2/2005).
- SourceForge.Net. Miranda Instant Messenger. <http://miranda-im.org/>, 2005b. (Accessed 22/2/2005).
- Watzlawick, P., Bavelas, J.B. and Jackson, D.D. Pragmatics of Human Communication: A Study of Interactional Patterns, Pathologies, and Paradoxes. W. W. Norton and Co., N.Y., 1967.
- Wikipedia. Instant Messenger - Wikipedia, the Free Encyclopedia. http://en.wikipedia.org/wiki/Instant_messaging, 2005. (Accessed 18/2/2005).

Contact

Prof. Dr. Tom Gross, [tom.gross\(at\)medien.uni-weimar.de](mailto:tom.gross(at)medien.uni-weimar.de), T. 03643/58-3733
Christoph Oemig, MSc., [christoph.oemig\(at\)medien.uni-weimar.de](mailto:christoph.oemig(at)medien.uni-weimar.de), T. 03543/58-3773

Acknowledgments

We thank our colleague Andreas Ziemann for inspiring discussions on identity and the works of Goffman, our colleague Tareg Egla for providing the Sens-ation platform, and our Cooperative Media Lab (CML) students Andre Kunert, Kai Riege, Robert Gerling, Yunlu Ai, Andrea Lahn, Nicolai Marquardt, Christian Semisch, and Matthias Pfaff for contributing to the concepts and implementing the platforms as well as the anonymous reviewers for valuable comments for improvements. und Axel Sylvester und Unterstützung der im/ve group am studentischen Projekt "ElMusicCo" im WS 2004/2005