

Smart Instant Messenger in Pervasive Computing Environments¹

Chun-Fai Law, Xiaolei Zhang, Sung-Ming Chan, Cho-Li Wang

Department of Computer Science,
The University of Hong Kong,
Pokfulam Road, Hong Kong

Abstract. In this paper, we explore the potential of extrapolating the instant messaging paradigm into pervasive computing environments. Under this vision, an instant messenger is regarded as a unified interface for all communications among human, software services and various devices. To meet the demands, we introduce a novel instant messenger system i.e., *Smart Instant Messenger*, with original features of context-aware presence management, dynamic grouping, and resource buddy services. This system is built atop a context-aware supporting middleware, which adopts an ontology-based context model and handles the chore of retrieving and managing context information. Jabber protocol is exploited as the underlying message exchange format for extensibility. The system prototype is implemented and evaluated with respect to the responsiveness of queries and memory usage of the middleware.

1 Introduction

Instant messaging (IM) has been booming since its birth and gradually becoming the most popular communication tool [15]. IM is characteristic of instantaneous message delivery and presence awareness. In particular, presence awareness differentiates IM from other communication paradigms. We believe such features fit naturally into pervasive computing environments, where communication and awareness are essential. Under this vision, “chat” would no longer be the privilege of human; rather, interactions between human-software, software-device and device-device could freely take place. We envision the potential of extrapolating IM paradigm as a unified interface for all communications. Aiming this, we have identified several new design concepts including context-aware presence management, resource buddy services and dynamic grouping.

Presence information shows a user’s responsive status, i.e., availability to be involved in a conversation. Current IM products predefine a set of options such as online, busy, and away. This coarse-grained categorization of user status, however, is incompetent under the pervasive vision. We propose a context-aware presence management approach and introduce improvements from three aspects:

¹ This research is supported in part by a CERG grant (HKU 7146/04E) from the Hong Kong Government.

(1) Context should be used as presence information. Apparently, when a user is aware of the other's situation such as her location, activity, security level and mood, they could communicate more appropriately. An imperative case is the mobile IM system, where showing "online" is meaningless if the user just keeps the connected device in pocket. (2) Presence information should be disseminated in a context-aware manner. Current IM products show the same status of a user to all her buddies. In reality, however, a user's availability is affected not only by her own situation, but also by the relationship with the corresponding buddy. For example, we ought to be "online" among the discussion members, yet appear "busy" to the outliers. (3) Presence information should be set automatically by the system. Nowadays an IM user needs to manually change her status, which tends to be burdensome and fallible. For a mobile user, things would be even more intractable, as her status might change frequently and in an arbitrary way. It is therefore appropriate for the system to handle this task, provided that the presence information can be automatically induced.

In pervasive computing environments, all smart artifacts can "talk" with you. Should they each adopt their own "dialects", a human user would be obliged to master a multitude and burdened in shifting the "language" to and fro. Also, it would involve a great deal of human attention to monitor, control and utilize various resources. Reflecting on the success of IM, we borrow the idea of "buddy" and view human, software and all sorts of devices uniformly as parties of communication. We also propose to use IM as the unified interface. Via IM, a user may include all usable resources in her contact list and "talk" with them in a personalized way. Another advantage of this approach is that, the user and the resource buddies could mutually stay aware of each other's status. The user can quickly tell which resource is near and ready for use, and select a "best" buddy to serve her purpose. Vice versa, the resource buddies could observe the user's situation and decide on the most appropriate way to interact with the user. For example, a notification service could choose to call the user's office phone if she is there, or email a reminder if she is temporarily away.

Grouping mechanism is commonly adopted in IM products to organize the buddy list for the user. In current situation, strategies for grouping are typically framed by the producer and remain unchanged after distribution. Groups are set by the user once for all. In real life scenarios, however, human relationships might be temporary, impromptu and varying. We devise a novel dynamic grouping mechanism so that: (1) Grouping should be adaptive i.e., able to change automatically according to the real situation; (2) Grouping should pertain to the user's requirement. For example, grouping the buddies by their locations can help a user "bump into" an acquaintance in a crowded hall, and grouping the relevant members of the same task can speed up the collaboration efficiency.

In this paper, we present our Smart Instant Messenger (SIM) system which fulfills the new concepts listed above. Section 2 overviews the SIM system design, elaborates on how the new features are realized and then introduces the context-aware supporting middleware, which underlies the SIM framework. Details of system implementation and experimental results are given in Section 3, followed

The second type of presence information embodies the subset of a user's context which she is willing to disclose, including for example her current activity, location and the people nearby. This is enabled by CASM, which actively collects user context on behalf of applications. When a user's context is inquired by a buddy as presence information, her context is encapsulated in an XML-formatted message, routed to the buddy's client, parsed and displayed on the roster.

In SIM, human users and resource buddies are conceptually identical. One slight difference is that, upon initialization, the resource client uses its resource module to register to the local SIM server, while the user client performs service discovery on whenever necessary. There are two ways to communicate with the resource buddy. One is to use the original chat window and type the user-defined commands; the other is to download a UI from the resource, which is described in an XML DataForm format. According to devices' configurations and users' preferences, the UI may be rendered in different customized ways.

The SIM system provides an extensible set of grouping mechanisms, including location, activity, hobby and relationship. In the current stage, we specifically investigate the location-based grouping and activity-based grouping. Location-based grouping retrieves from CASM the most updated user locations and groups the buddies of the same location. This is especially useful to help a mobile user, when entering a place, to "bump into" an acquaintance and to initiate a serendipitous interaction. It will also keep the user informed of the surrounding resources. Activity-based grouping reflects on the "distraction-free" tenet, aiming at facilitating the user's activity (or task) by grouping the relevant people, materials and resources together. For example, suppose a user is involved in preparing a project presentation, SIM will dynamically group the project members, documents, applications, printer and projector in her buddy list, forming a virtual collaboration environment, so that she could easily reach what she needs to contact or utilize. Current implementation assumes the user's activity can be inferred and the relevant information are stored in the context knowledge base. Upon request, CASM will retrieve the information of all possible buddies (human as well as resources) and return the result to the roster module in the SIM client, which in turn updates the grouping.

2.3 The Context-aware Supporting Middleware

The SIM system explicitly separates the context processing routines from application logic. A generic context-aware supporting middleware (CASM) handles the chore of processing, interpreting and reasoning over context information retrieved from various context providers. This separation principle not only relieves the burden of context-aware application programmers, but also fosters the reuse of context and context reasoning processes.

CASM centers an ontology-based context model for a formal context representation, which facilitates knowledge sharing in the open, heterogeneous pervasive environments, and enables various logic-based context reasoning mechanisms. Contexts are classified into five categories: Device, Person, Location, Time, and Activity. There are also relationship properties among these main

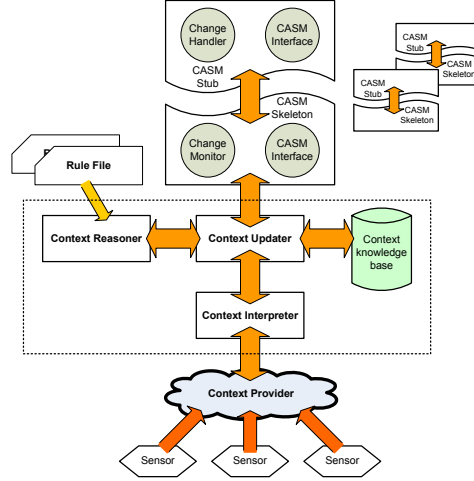


Fig. 3. Detailed design of context-aware middleware

classes. For example, an instance of class “Person” can have a relationship called “hasLocation” which links to an instance in the “Location” class. All classes and relationships can be added or removed as needed.

Figure 3 shows the detailed design of CASM. The *Context Interpreter* translates the context from heterogeneous sources to form an OWL instance data, which stores all the dynamic context information (e.g. location, time, current activity) in OWL files. The *Context Updater* directly manipulates the context model. When the context model is first created, the schema file will be parsed and data type of the domain and range for each property are specified. The *Context Updater* validates the data type for that particular context statement each time an add/remove request is received. Upon a context query, it inquires the context model and formulates the answer in a regular format that can be used by the client side easily. The *Context Reasoner* provides two kinds of reasoning over the context ontology, i.e. the transitive reasoning and the rule-based reasoning. The former is used to store and traverse class and property lattices. The latter supports user-defined rule set. Depends on the schema and domain of the ontology bound to the context model, rules can be written to derive the existence of some implicit information or map information to a standard format for applications.

CASM also provides a set of standard methods for application developers to update, query and register context event listeners to the middleware. An application registers interested context events to CASM, and relies on the latter to monitor the environment on its behalf. Notifications will be fired when the events happen, and the *Change handler* module in the application will invoke the corresponding event handling methods.

3 Implementation and Experiments

We have implemented SIM server and two versions (PC and PDA) of SIM clients. The SIM server uses and extends the Jabber open source server. We extends Jabber's Extensible Messaging and Presence Protocol (XMPP), which is currently an Internet Engineering TAsk Force draft, to report the state of buddies and to handle the interaction among human, software and devices through XML messages. SIM clients modify the open source Jabber client program "JBother" [9] to introduce context-aware presence management, resource buddy and dynamic grouping. Figure 4 shows the client-side GUIs running on the PDA (HP iPAQ H5500). Figure 4 (a) shows the message dialog, Figure 4 (b) illustrates the SIM's roster, which groups the buddies relating to the SIM project, including 6 members and a printer. Figure 4 (c) shows the location-based grouping, where the buddies are organized under the groups of canteen, lab and office.

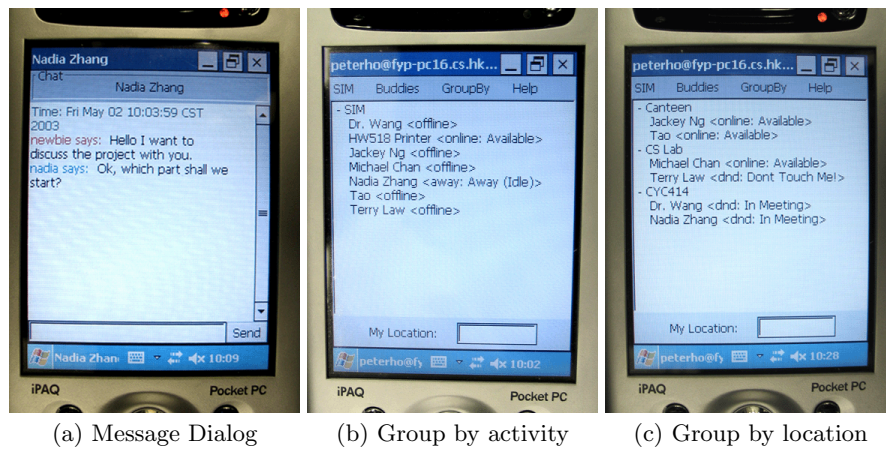


Fig. 4. Client-side GUI

We built up several ontologies for pervasive computing environments. Figure 5 shows one ontology used for modeling the basic concepts of campus life. The Web Ontology Language (OWL) has been selected as the ontology language for its expressivity and standardization. Reasoning and inference over the context models are based on the Jena [10] framework. A set of rules has been developed to infer high-level context from low-level facts.

We notice the major time-consuming part of the system (wireless delay excluded) is related to operations on CASM middleware. As more context instances are added into the context knowledge base, the overhead of the middleware grows accordingly. To test the performance, we evaluate the responsiveness and memory consumption with the increase of the number of instances. The experiment proceeds as follows. A PC (Intel Pentium4 2.26GHz, 512MB memory, Linux FC 3.0) which runs Jena version 2.2. A typical sequence of operations is compiled

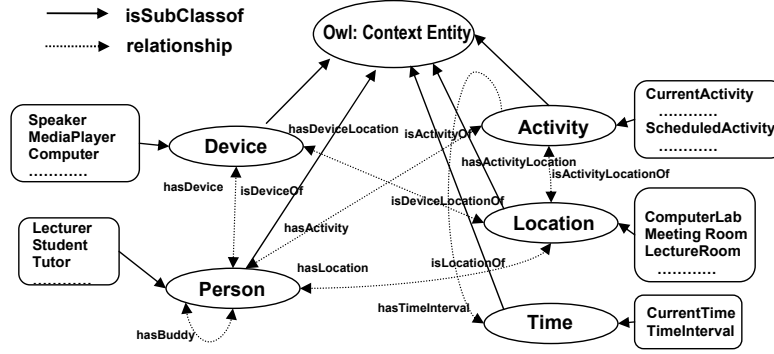


Fig. 5. Diagrammatic view of the campus ontology model

as a sample test, including 2 add (adding instance data into the ontology), 1 remove (removal of instance data), 1 class query and 1 instance query. We increase the instance at the number of 300, 700, 1000 and 1800. At each stage, the sampling sequence is performed and the total processing time and memory usage are measured. The result reports an approximately linear growth of memory usage varying from 17MB to 22MB and an average processing time of 3.4s with variations within 0.2s. The performance of the system is tolerable for non-crisis scenarios and the increase of instance will not cause much degradation.

4 Related Work

The idea of combining awareness with communication originates in computer supported cooperative work (CSCW) and human-computer interaction (HCI). Researchers in media space research [1][18] and awareness systems [3][6] have identified the importance of shared context to facilitate conversation. For example, social awareness has been explored in [7] and [14]. However, this stream of research mainly targets an efficient group collaboration among human users. Our work, on the other hand, considers all types of interactions including human, software and hardware resources.

The distinctive features of IM have been gaining more attention in recent years. Nardi [12] suggested that, beyond information exchange, IM could implicitly be used to negotiate availability, maintaining the sense of social connection and switching media. There've also been several research projects on extending IM with context-aware features. They could be broadly categorized into two groups. The first exploits context for conditional message delivery. For example, CybreMinder [2] allows users to associate the contextual information with to-do items and delivers them upon pre-defined condition. This can be viewed as a special type of context-aware one-way message delivery. Similarly, a handheld IM system described in [11] also empowers users to specify a set of situations that must be met before the system delivers the message. The second group explicitly uses context information to broaden the communication spectrum. ConChat

[16], for example, supports two conversational parties to exchange or query each other's context. The AwareNex [17] system from Sun Laboratories displays location and activity information of the users on the contact list. Similar to SIM, these projects emphasize more on the "outeraction" [12] functionality of IM. However, the issues of contextual presence and context-aware presence distribution are not sufficiently explored. Meanwhile, there tends to be little discussion on grouping mechanism, which we believe is also a tool of potential yet has been underused.

5 Discussion and Future Work

In this research, we have explored the vision of extrapolating instant messaging paradigm into the pervasive computing environments. We have designed and implemented the Smart Instant Messenger system, which transcends current IM products with new features including context-aware presence management, resource buddy services and dynamic grouping support. The system is built on top of a context-aware supporting middleware, which centers an ontology-based context modeling approach. Though at the prototype stage, it has already demonstrated the advantages for being used in pervasive computing environments. Experiments on performance evaluation also suggest its feasibility.

Our design fulfilled the following principles: *Separation of context provision from context consumption*. The chore of retrieving and managing context should not be directly integrated in an application; rather, a separate middleware layer or the systems infrastructure should be responsible for providing context information. SIM adopts a context-aware supporting middleware approach. It not only relieves the burdens of programmers and the small devices; the generic middleware could potentially support more applications.

Design for extensibility. Extensibility is essential in pervasive environments as users, applications, devices and sensors might all come and go dynamically. Also, the users' requirements might change over time. SIM chooses the Jabber protocol for its extensibility consideration, adopts a distributed architecture, and exploits an ontology-based context modeling solution to facilitate the re-use and integration of knowledge.

Prototype for real life usage. Pervasive computing is still in the germinal stage. We believe live applications will stimulate and inspire the research. Therefore this version of SIM is designed for using on campus, with resources, users and use cases rich enough for a real system.

We believe such an attempt is of great potential, both in practical usage and in research. Future work includes supporting user-level mobility of instant messenger among different devices, improving the performance of context-aware supporting middleware and exploring the SIM usage in hospital scenarios.

References

1. S. Bly, S. Harrison and S. Irwin. Media spaces: Bring people together in a video, audio and computing environment. *Communications of the ACM*, 36 (1), 28-46, 1993.

2. A. K. Dey and G. D. Abowd. CybreMinder: a context-aware system for supporting reminders. In 2nd International Symposium on Handheld and Ubiquitous Computing, volume 1927 of Lecture Notes in Computer Science, 172-186. Springer, 2000.
3. P. Dourish, S. Bly. Portholes: supporting awareness in a distributed work group. Proceedings of CHI'93 Human Factors in Computing Systems, 541-547, New York: ACM Press.
4. J. Fogarty, J. Lai and J. Christensen. Presence versus availability: the design and evaluation of a context-aware communication client. International Journal of Human-Computer Studies (IJHCS), Vol. 61, No. 3, September 2004, pp. 299-317.
5. D. Greene and D. O'Mahony. Instant messaging and presence management in mobile ad-hoc networks. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. March 14-17, Orlando, Florida, pp. 55-59, 2004.
6. C. Gutwin, S. Greenberg. Design for individuals, design for groups: Trade-offs between power and workspace awareness. In Proceedings of CSCW96 Conference on Computer Supported Cooperative Work, 207-216, New York: ACM Press.
7. S. E. Hudson and I. Smith. Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. Proc. Comp. Supported Cooperative Work, 1996, pp. 248-257.
8. Jabber Instant Messaging. Online resource. <http://www.jabber.org/>
9. JBother Homepage. Online resource. <http://www.jbother.org/>
10. Jena: a semantic Web framework for Java. Online resource. <http://jena.sourceforge.net/>
11. Miguel A. Munoz, Marcela Rodriguez, Jesus Favela, Ana I. Martinez-Garcia, Victor M. Gonzalez. Context-aware mobile communication in hospitals. IEEE Computer, vol. 36, no. 9, pp. 38-46, Sept 2003.
12. B. Nardi, S. Whittaker, E. Bradner. Interaction and outeraction: instant messaging in action. In Proceedings of ACM 2000 Conference on Computer Supported Cooperative Work, 2000.
13. A.J.H. Peddemors, M.M. Lankhorst, J. de Heer. Presence, location and instant messaging in a context-aware application framework. 4th International Conference on Mobile Data Management (MDM2003), Melbourne, Australia, Jan 2003.
14. E. R. Pedersen and T. Sokoler. AROMA: abstract representation of presence supporting mutual awareness. Proc. SIGCHI Conf. Human Factors in Comp. Sys., Atlanta, GA, Mar. 22-27, 1997, pp. 51-58.
15. Pew Internet & American Life Project. How americans use instant messaging, Sept 2004. Online resource. http://www.pewinternet.org/pdfs/PIP_Instantmessage_Report.pdf
16. A.Ranganathan, Roy H. Campbell, A. Ravi, and A. Mahajan. ConChat: a context-aware chat program. Pervasive Computing, 1(3):51-57, July-September 2002.
17. J.Tang, N. Yankelovich et al. ConNexus to AwareNex: extending awareness to mobile users. Proc. SIGCHI Conf. Human Factors in Comp. Sys., Apr. 1998, pp. 566-73.
18. S. Whittaker, G. Swanson, J. Kucan and C. Sidner. Telenotes: managing lightweight interactions in the desktop. Transactions on Computer Human Interaction, 4(2):137-168, 1997.