RC22684 (W0212-131) December 20, 2002
Computer Science

IBM Research Report

CoPlace Defined: A Service-based View of Online Places

John Patterson, Li-Te Cheng, Werner Geyer
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

CoPlace Defined:
A Service-Based View of Online Places

John Patterson, Li-Te Cheng, Werner Geyer

IBM T.J. Watson Research, One Rogers Street,
Cambridge, MA 02142 USA
{john_patterson, li-te_cheng, werner.geyer}@us.ibm.com

Abstract. This paper investigates the relationship between online places and Web Services. While the metaphor of online places has been useful for many people in their effort to comprehend the World-Wide Web, it is not sufficient to define a Web Service for online Places. In an effort to define such a Web Service, we outline the requirements of a family of Web Services. We are striving towards the definition of a so-called CoPlace, which is an online place built by combining some elementary place services.

1 Introduction

Denizens of the Internet speak of “going to” eBay or Yahoo, as if they are about to take a bus down the street. A web site becomes a place to visit. It contains the things you want. At some places you can even meet your friends. The properties of real places -- location, containment, presence, and context -- all map onto similar online experiences. We see it in common parlance, in our technical literature, and in our marketing brochures; each hopes to impart the connotations of physical place to online systems. The metaphor has moved beyond a useful intellectual tool; it has become essential to most people’s understanding of the Internet.

Online place is a metaphor, however, not a definition [11]. Many quite distinct online experiences are considered places, such as web sites, chat rooms, discussion areas, collaborative virtual environments (CVEs) [3], and electronic meetings. While serviceable for conversation and conducive to creativity, the term place is simply not a technological construct.

Nevertheless, as the Internet moves towards a component-based, service-oriented model that relies on the emerging concept of Web Services as the building blocks of future Internet applications [15], the concept of place will need a more precise definition. These Web Services define standard communication patterns among computers to implement Internet applications. Among the benefits of this standardization are code reuse, separation of User Interface concerns from data concerns, and the opportunity to federate access to similar service providers.

In this paper, we address the relationship between Web Services and online or “virtual” places. The first objective of our work is to move towards the development of Web Services for online places. To develop these Web Services, we must begin the hard task of defining the properties of places that will and will not be incorporated into the services. While the place metaphor remains a guide, it is not sufficient.

The second objective of this paper is to ensure that our definition of an online place achieves a unification of the two main types of places, i.e., asynchronous and synchronous places [12]. In asynchronous places, users interact via the messages or objects deposited in the place. In synchronous places, users feel as if they interact directly in real-time with one another and with the objects that they jointly observe.

We feel that users are not especially interested in this distinction. They would be happy if places behaved asynchronously when no one else is around and synchronously when multiple users are present. Research has long recognized the importance of seamlessness in collaborative systems [1]. We believe that a Web Service for online places should strive for the ideal of blending the elements of asynchronous and synchronous places.

With this end in mind, we have coined the term *CoPlace*[1] to signify an online place that combines both synchronous and asynchronous capabilities. Thus, our objective is to define the properties of a Web Service for a CoPlace.

Figure 1 illustrates the strategy we are adopting. We start from the notion of a *Base Place* that provides a persistent object store for shared information. This Base Place is sufficient to characterize a web site as a place, but does not capture some of the more advanced properties of a place.

From the Base Place we derive three specializations: C*onversational Place*, *Synchronous Place*, and *Rendezvous Place*. Each assumes the Base Place capabilities and adds additional requirements that qualify it as an advanced service. We call these *component places* because we consider them as composable services from which different types of places can be built. We then define a CoPlace as a service that possesses properties of all three component places.

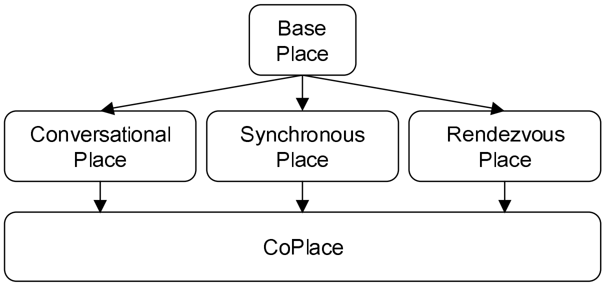

Fig. 1. Place map

Our paper is structured according to this categorization. For each type of place we define mandatory requirements but also other optional properties. Optional properties of one type of place might be mandatory for another type of place. We provide canonical examples of systems that match the criteria. In addition, we consider a

[1] CoPlace is derived from a contraction of Collaborative Place. We will use the new term in the remainder of this paper.

number of examples that we feel do not conform to the requirements of each type of place. Our objective is to understand, generally, what a Web Service for each type of place might be like; we are not interested in specifying the protocol details.

2 The Base Place

In this framework, the Base Place captures the service requirements of every place, since all places are built upon it. We have designed it so that most web sites could be considered a Base Place. The aspects of the place metaphor captured in the Base Place definition are:

- location,
- object containment,
- permanence, and
- commonality of experience.

In more familiar terms, a Base Place provides an enduring location for things that are experienced in the same way by those who go there.

2.1 Base Place Requirements

A Base Place is a persistent, multi-user, online object store. A Web Service for a Base place must exhibit the following properties:

Addressability: A user can access a Base Place using a single, consistent naming or addressing scheme, regardless of the client software implementation or the user's actual online location. In practice, this usually means the Base Place has a known network address. The addressability property gives the place a location.

Containment: The Base Place acts as a container for a finite population (that could increase or decrease over time) of objects. Note that objects may contain other objects. The containment property defines the boundaries of the Base Place. Often the content of the objects contained by a Base Place characterizes the purpose of the place.

Persistence: Some objects contained by the Base Place will continue to exist during the lifetime of the Base Place unless explicitly deleted. These persistent objects provide the sense of permanence and context to the Place. The Base Place itself is expected to outlast the coming and going of its users and continue to exist until destroyed by the Base Place's administrator (a user who manages places).

Multi-user Readability: At least two users can read the same nonempty set of objects in the Base Place at any time. Moreover, although each user's experience of the objects might vary in some degree, the retrieved object values should be essentially similar. Without this requirement each user's experience of the site could be entirely different and indistinguishable from visiting entirely different sites.

2.2 Extensions and Examples

We view these four properties: addressability, containment, persistence, and multi-user readability, as requirements, but there are many other optional properties to consider. For example, we anticipate that objects might also be writable, deletable, searchable, etc. Moreover, we expect that users and objects may have varying *permissions*. For instance, a user with administration permissions could read, write, edit, delete move, search, or rename any objects in the base place. To be a Base Place, however, the only access requirement is multi-user readability.

While some of the more interesting opportunities for extending the Base Place are discussed in later sections, we cannot address all the optional opportunities for refinement in this paper. Instead, let’s ask what types of online activities are excluded under the current definition of a Base Place.

Is the World-Wide Web as a whole a Base Place? According to our definition we would have to say, no. There is no particular address that corresponds to the whole web. Moreover, it is questionable whether the notion of containment should be permitted to include the set of everything.

Is an online conference a Base Place? Again, for most such conferences, we are inclined to say no. In some cases, there is no address for the conference; participants invite other participants in, but there is no real sense of location. Even though some conferences have a published address, they often lack any persistence. The objects built during the conference disappear at the end of the conference. Indeed, the conference itself has no permanence; its address and contents go away when the last participant leaves.

Is an online database a Base Place? Perhaps, but to qualify it must support data objects that can be seen by multiple users. Consider a database that keeps track of each individual’s private information, which may only be viewed by that user. It has an address, containment, and persistence, but since no two people ever have access to the same content it does not qualify as a Base Place.

Finally, let’s ask a particularly difficult question. Is email a Base Place? We can make it a little easier by restricting our attention to email that is addressed to a mail group. There is an address; there is persistence; and there is multi-user readability. Only containment seems questionable. What makes it especially peculiar is the distributed nature of the objects and the wholly individualized control over deletion. Nevertheless, if the users of a mail group never deleted any of the messages addressed to the mail group, then their experience of that mail group might be indistinguishable from a more centralized implementation of a place. At present, we are not inclined to change our definition in light of this email example; we simply note it as a peculiar consequence of our current approach.

3 Conversational Place

Our first specialization of the Base Place is a Conversational Place. While the Base Place requires the ability for multiple users to experience the place in a similar manner, it does not require any ability for the multiple users to interact with one another using the place. The Conversational Place addresses this issue. It is a Base Place that supports the notion of interaction or conversation among its users.

Discussion forums such as a Usenet newsgroup and most Lotus Notes applications [13] are familiar examples of Conversational Places. Everyone can see what's going on and add to the conversation as needed.

3.1 Conversational Place Requirements

The conversation revolves around the changing state of one or more objects shared amongst two or more users. More precisely, a Web Service for a Conversational Place has all the properties of the Base Place and must exhibit the following additional properties:

Conversational objects: There must be a nonempty subset of the shared objects that are used for conversation.

Conversational users: There must be two or more users who have the ability to interact with the conversational objects.

Common read access: The conversational users must have read access to the conversational objects.

Modification access: Each conversational user must have modification or creation rights on some of the conversational objects.

3.2 Extensions and Examples

While this definition ensures that a group of users can experience each other's actions it is missing two properties that are tempting to include. First, there is no provision for knowing which user has acted on the conversational objects when they are modified. A Conversational Place may optionally preclude anonymity, but this is not required.

Second, it is tempting to impose some requirement for revealing the order of the conversational actions. For example, most discussion forums provide a means to respond to an earlier posting, which preserves the relationship among the objects and reveals the order of the actions. We treat this as an optional capability, not a requirement even though it opens the possibility of Conversational Places with incoherent or, at least, difficult conversations. We have not attempted to preclude such examples because the requirements are likely to be quite complicated. It is an area that deserves further consideration, however.

As before, we can illustrate the requirements with some challenging cases. For instance, is email that is sent to a mail group a Conversational Place? If email to a group can be a Base Place, then it can probably be a Conversational Place as well. Everyone in the group has read access to the messages and everyone may author a new message for all to see.

Is an online game a Conversational Place? Unlike discussion forums, online games do not offer tidy back and forth messages. They have objects controlled by each user that are continually modified and a set of game rules that lead to other state modifications as well. Nevertheless, the state is generally viewable by all and each user has some ability to affect the state. Conversation might be the last thing on the players' minds, but we would view online games as a Conversational Place. Note that many games today also possess synchronous properties in addition to the properties of a Conversational Place.

Is a FAQ (Frequently Asked Questions) or moderated discussion a Conversational Place? These systems allow all users to submit messages, but only a few to respond. By our definition, this would be a Conversational Place, since all may see what's posted and anyone may post something. Nevertheless, it seems not altogether in the spirit of conversation.

Is a suggestion box a Conversational Place? Since a suggestion box usually allows one to create objects, but not to see the objects that others create, we are inclined to say no. It might be a Conversational Place for the employees who must read and sort through the suggestions, but it is usually not a Conversational Place for those who submit suggestions.

4 Synchronous Place

Our definition of a Base Place provides no mechanism for users to be informed when something interesting happens at the place. They can see what's happening when they ask, but they cannot simply be told when it happens. Some databases offer a triggering capability that will send a notification to a user when some event occurs. Assuming the database qualifies as a Base Place and the trigger is timely enough, the introduction of the triggering facility makes it a Synchronous Place.

4.1 Synchronous Place Requirements

The Web Service for a Synchronous Place has all the properties of a Base Place and must also exhibit the following properties:

Subscription: A Synchronous Place must provide at least one type of *event* to which users may subscribe. A subscription is an object that specifies a user's interest in events occurring in the place. Subscriptions may be persistent or transient depending on whether they must endure beyond the most recent use of the place or survive a server reboot. Users may register multiple subscriptions. The combined set of subscribed events at any time is called an *interest profile*. Events may be of any sort, but are usually actions on objects within the place (e.g. the creation of a new object, a series of edit operations, etc). A subscription may be created explicitly due to a user request, or implicitly as a consequence of some user action, such as logging into the place.

Notification: Notification is an action taken automatically by the place, consisting of a *condition* and an *alert*. The *condition* is the trigger when some event in the place matches a subscription's event profile. The *alert* is the delivery of a message characterizing the event to the subscribing user. Notification allows the place itself to interact with the user.

Appearance of simultaneity: Notifications are timely. That is, events are delivered with only a small lag between the occurrence of the event and the receipt of the notification. The definition of a small lag will vary with applications, but is generally interpreted to mean small relative to human perception and cognition.

4.2 Extensions and Examples

If achieving a small time delay were not critical, then we might consider a number of approaches to transmitting notifications. For example, a web site with selective dissemination of information might use email to deliver its notifications of new articles. This is sensible given the needs of the user and the fact that the notification is a presentation directly to the user. Using email for notifications is somewhat more problematic when the goal is to deliver the notification to a program and not the user. Nevertheless, a creative use of mime types and mail agents can ensure that email is a viable transport mechanism for less urgent notifications.

On the web a common approach to obtaining notifications is to have the Place accumulate events and then deliver them when asked. This is referred to as polling the server. Since the Hypertext Transfer Protocol (HTTP) expects a query from the client for each response given by the server, polling is a technique for simulating notifications. It is an effective technique assuming the temporal demands are not too high. Polling on the order of minutes is workable, though potentially wasteful of network and computing resources.

These two approaches to notification are useful in many applications, but the challenge posed by a Synchronous Place is to impart a sense of simultaneity to the user. This is the essential requirement of synchronous groupware. For example, if the object being monitored is a ticker tape listing stock prices, some users have an interest in knowing about changes immediately. Email delivery of notifications is definitely too lazy and polling on the order of every second is too resource intensive. Supporting the appearance of simultaneity requires something other than HTTP in its current form.

Interesting examples of Notification Places providing the appearance of simultaneity are Linda [9], JavaSpaces™ [8] and Gryphon [2]. Some versions of these systems support persistent objects and a scheme for learning when the objects change. Arguably, the Web Service advocated here would be a platform-neutral approach to implementing these systems.

The Notification Service Transfer Protocol (NSTP) [5], appropriately modified, could be considered a precursor to the Web Service anticipated here. That protocol had implicit subscriptions only and the state was considered transient rather than persistent, but it was designed to be a platform-neutral Internet protocol for issuing notifications about state changes.

5 Rendezvous Place

None of the places discussed so far requires the identification of users or their actions. We have not precluded associating user names with the objects they modify, but we have not required it either. With the Rendezvous Place we introduce the ability for users to see who else is present in the place. Consider, for example, a web site that provides a list of all the visitors who are currently at the site. This is a clear example of turning a simple Base Place into a Rendezvous Place.

5.1 Rendezvous Place Requirements

The Web Service for a Rendezvous Place has all the properties of a Base Place and must exhibit the following additional properties:

Identifiable Users: Whether established by some secure authentication technique or simply generated by the user, each user must be associated with a name or token that may be used to identify that user to the other users. While it is desirable to enforce uniqueness, this need not be a requirement of the service.

Visit: A user must be provided some means to associate with and disassociate from the place. The period during which the user is associated with the place is called a visit. The action that explicitly establishes the association is referred to as "entering" the place. This is usually when the user provides his/her identifying token. Thereafter, the visit continues until either a service error causes the visit to terminate or the user explicitly "leaves" the place.

Presence Objects: During the period of a visit, a presence object is created at the place for each visiting user. Presence objects are usually implemented as transient objects, created when a user "enters", and destroyed when the user "leaves". Persistent implementations are possible, however. For example, one might wish to log a user's comings and goings in a persistent fashion and use the most recent entry as the active presence object. While the presence objects are only required to indicate a user's presence or absence, they may be used to indicate other aspects of a user's availability, such as whether they are willing to be disturbed, or where they may be contacted.

Read Access to the Presence Objects: The Rendezvous Place must provide read access to the Presence objects.

5.2 Extensions and Examples

The pure case of a Rendezvous Place is a web site that keeps track of who is visiting at any moment and presents that information on every page. The users identify themselves and enter the Place on the first page visited. Leaving could be managed by a combination of a timeout and a keep-alive request from within every page.

This approach has the virtue of building on HTTP alone, but it poses some interesting problems. Unless there is polling to maintain synchronization, users may appear to be present after they have moved on to a different web site. Also, until one refreshes a page, users who have recently arrived may not yet be seen. For these reasons, a Rendezvous Place is usually also a Synchronous Place (discussed below). The subscriptions are often implicit with a visit implying a subscription to all presence changes until the visit ends. This approach and policy is not the only one, however. Subscriptions need not require the subscriber's presence to be active. Also, explicit subscriptions are especially useful when a very large number of users visit the place.

A challenging question is whether general online awareness implies a Rendezvous Place. The Buddy list capability implies a visit (of sorts), presence objects, and subscriptions for presence. But, what is the Base Place? Is it the whole World-Wide Web? Is there really an address or object containment? At present, we are inclined to view online awareness as something other than place-based because it lacks the clear context conveyed by a Base Place.

6 Permutations and the CoPlace

The previous sections specify the minimal requirements for a Base Place and three specializations, the Conversational Place, the Synchronous Place, and the Rendezvous Place. Our intention is that the three specializations should be composable, meaning that new place types may be derived through combining the capabilities of each. Thus, we can have a Synchronous-Rendezvous Place and a Conversational-Rendezvous Place and so on. We refer to the type of Place that meets all requirements as a CoPlace. In this section, we start by describing the permutations of these places and the consequences on their capabilities. We provide examples of the ultimate combination - the CoPlace - and we end by considering some requirements that relate to ensuring that the separate Place types are composable.

6.1 Composed Places

The requirements of each type of Place postulate a certain class of objects at the Place and a certain type of capability with respect to this or other classes of objects. For example, the Base Place refers only to generic objects and requires only multi-user readability of these objects. This is sufficient to create a commonality of user experience, but little more. To achieve interaction at a Place, we introduce conversational objects and require that they be modifiable by some user and readable by all. To achieve a sense of simultaneity, we introduce subscription objects that give rise to a new type of access, i.e., notifications. Finally, to achieve a sense of being there, we introduced presence objects readable by all.

What makes the composed places interesting is the manner in which the capabilities of each are applied to the objects of the other. For example, the combination of the Conversational and Synchronous Places can yield two interesting consequences. The more familiar capability is to permit subscriptions to conversational objects. This yields any of a wide variety of synchronous artifacts, e.g., a chat space, a shared white board, or a multi-user game. An example of a Conversational-Synchronous Place is a shared central database. The database's users enter the shared central database but need not be visible to each other. The users engage in "conversation" by reading and modifying database documents. The users are notified immediately of changes to their documents or conflicts (e.g. two users may be modifying the same document).

A less obvious implication of this composed place occurs if the subscriptions are treated as conversational objects. This might mean that my subscriptions are available to everyone, but it also might mean that others may modify my subscriptions. Although subscriptions are not usually treated as anything other than the creator's business, this could be a way for users to either understand or compel the attention of one another.

The combination of the Conversational and Rendezvous Places allows a semi-synchronous conversation among users. The users may learn who is present as they look at or create documents. The "Conversation" would probably become a more immediate discussion focused among those users present rather than the extended exchange of correspondence in a newsgroup-like setting. However, this is not a truly synchronous; there is no timely notification of presence. Thus, users are either forced to poll the presence objects or settle for awareness that is no more accurate than the last page retrieval. The result is a purely HTTP-based chat forum – a web page where users can log in, chat, and log out. An extreme case is a website like Amazon or eBay, where customers login, post their reviews of products, leave some residue of their presence via Amazon's collaborative filters, and are automatically logged out when the web browser is closed.

The combination of the Synchronous and Rendezvous Places is a purely presence-driven Place. Since the only objects that are guaranteed to change are the presence objects, these are the only objects for which a subscription is likely to be interesting. Users manifest themselves to others via timely notifications from events related to each others' presence objects (e.g. entering and leaving the Place).

6.2 The CoPlace

The ultimate combination of all three specialized Places, is the CoPlace. From the Base Place it obtains the basic features of the Place metaphor: location, containment of objects, permanence, and commonality of experience. From the specialized Places it obtains user interactions, simultaneity, and presence. Conversations and presence can be reflected immediately to the users. Users may interact via the conversational objects regardless of whether anyone else is present in the place, but if someone is present, they may know immediately of any changes that are occurring.

A few examples of CoPlaces include MUD/MOOs [4], Collaborative Virtual Environments (CVEs) [3], Lotus QuickPlace [14], Babble [7], Loops [6], and Groove [10]. These systems provide a shared place where users can visit, converse, and be notified. They differ in implementation, goals, and more notably, in the extent of their functions, but all provide an enduring place (Base Place) in which users may interact synchronously (or asynchronously) and be aware of one another's presence.

The Babble system is an interesting instructive example because of its simplicity. The conversational objects are the chat entries, which are stored persistently to ensure context to those who arrive late. Indications of presence are continually updated for all users.

6.3 The requirements of composition

Composition of the Web Services for the three specialized types of places will not come for free. It helps that all are derived from a common underlying approach, the Base Place. Nevertheless, each Web Service will need to be designed with composition in mind. The requirements stated so far do not address these sorts of issues.

A common representation of objects across the Web Services will be essential. This should include a common scheme for naming objects, events, actions, and capabilities. Without this, how can a subscription refer to conversational or presence objects or the events that give rise to a notification?

Another class of composition problem relates to the manner in which an object in one service might be valuable for controlling the behavior of another. For example, a common policy is that you cannot receive notifications unless you are present. In this case, the presence object is controlling the behavior of the notification service. Similarly, you might not be able to converse without being present.

7 Conclusion

Our goal here has been to suggest requirements for Web Services designed to capture the notion of an online Place. We have started by identifying a Base Place from which we claim that all other places derive. Then we identified three specialized extensions of the Base Place: the Conversational Place, the Synchronous Place, and the Rendezvous Place, Finally, we considered how these specializations might compose together to form a CoPlace comprised of the capabilities of all three specializations.

Table 1 summarizes some of the requirements of each type of place in terms of the major object type each introduces and the access capabilities that are needed. An example is provided to illustrate the type of place. The Base Place models a web site and requires little more than commonality of user experience. The Conversational Place models a newsgroup and ensures the ability to affect the place's state in a manner that others can see. The Synchronous Place supports the appearance of simultaneity by introducing a new type of access using notifications. And, the Rendezvous Place allows users to know of one another's presence at the Place.

The combination of all three types of places yields what we call a CoPlace. Examples of CoPlaces include MUDs/MOOs, some types of chat rooms, Collaborative Virtual Environments, and other synchronous groupware systems with persistent capabilities. We believe that a CoPlace offers an effective blending of the two major types of groupware: asynchronous and synchronous.

Ultimately, the hope is that bringing all these otherwise disparate systems into a common framework will make it easier for developers to start building upon one another's accomplishments. A well-designed Web Service for accessing CoPlace's would permit the implementation of the CoPlace client to become more independent of any CoPlace server implementation. Creative uses of the CoPlace could be developed that move beyond the original intention. For example, someone might implement a CoPlace browser that builds upon and unifies the input from many CoPlaces.

Table 1. The requirements of the Base Place and the three specialized Places.

Place Type **(Example)**	**Requirements**	
	Objects	**Access**
Base (e.g. web site)	Generic Persistent Objects (a web page)	Multi-user readability
Conversational (e.g. newsgroup)	Conversational Objects (a news posting)	Common read access
		Modification access
Synchronous (e.g. Linda/JavaSpaces)	Subscription (list of topics of interest)	Notification
Rendezvous (e.g. web based in/out board requiring manual refresh)	Presence Objects (an online-status icon)	Presence read access

8 Acknowledgements

We thank the members of the Collaborative User Experience (CUE) research group for their help and encouragement in developing these ideas.

References

1. Borghoff, S., Schlichter, J.H. Computer-Supported Cooperative Work: Introduction to Distributed Applications, Springer, Berlin, 2000.
2. Banavar, G., Chandra, T., Mukherjee, B., and Nagarajarao, J., An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems, Proceedings of the International Conference on Distributed Computing Systems, Austin, Texas, IEEE Computer Society Press, 1999.
3. Churchill, E. F., Snowdon, D. N., Munro, A. J. Collaborative Virtual Environments: Digital Places and Spaces for CSCW, in *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*, Churchill, Snowdon, Munro (Editors), Springer, London, 2001, 3-17.
4. Ciskowski, D., Benedikt, *C. MUDs: Exploring Virtual Worlds on the Internet*, Macmillan Computer Publishing (Brady Imprint), May 1995.
5. Day, M., Patterson, J., Mitchell, D., The Notification Service Transfer Protocol (NSTP): Infrastructure for Synchronous Groupware, in *Proceedings of the Sixth International World-Wide Web Conference*, April 7-11, 1997.
6. Erickson, T., Kellogg, W. Social Translucence: An Approach to Designing Systems that Mesh with Social Processes, in *Transactions on Computer-Human Interaction*, Vol. 7, No. 1, ACM Press, New York, 2000, 59-83.
7. Erickson, T., Smith, D. N., Kellog, W. A., La, M., Richards, J. T., Bradner, E. (1999) Socially translucent systems: Social proxies, persistent conversation, and the design of Babble, in *Proceedings CHI '99, Human Factors in Computing Systems*, Philadelphia, PA, 1999.
8. Freeman, E., Hupfer, S., Arnold, K. *JavaSpaces™ Principles, Patterns, and Practice*, SUN The Java Series, Addison Wesley Professional, 1999.
9. Gelernter, D., Carriero, N., Chandran, S., Chang, S. Parallel programming in Linda, in *Proceedings International Conference on Parallel Processing*, August 1985, pp. 255-263.
10. Groove Home Page, http://www.groove.net/, 2002.
11. Harrison, S., Dourish, P. Re-Placing Space: The Roles of Place and Space in Collaborative Systems, in *Proceedings of CSCW'96*, Boston, MA, ACM Press, 1996, 67-76.
12. Johanson-Lenz, P., Johanson-Lenz, T. System Design, in *Computer Mediated Communication Systems: Status and Evaluation*, Kerr, E.B., Hiltz, S.R. (Editors), Academic Press, New York, 1988.
13. Lotus Notes Home Page, http://www.lotus.com/home.nsf/welcome/lotusnotes, 2002.
14. QuickPlace Home Page, http://www.lotus.com/quickplace, 2002.
15. W3C, Web Services Activity, http://www.w3.org/2002/ws/, January 2002.