R-U-In? – Exploiting Rich Presence and Converged Communications for Next-generation Activity-Oriented Social Networking

Nilanjan Banerjee*, Dipanjan Chakraborty*, Koustuv Dasgupta*, Sumit Mittal*, Seema Nagar*, Saguna†
*IBM Research, India Research Lab, New Delhi, India
†Department of Information Technology, Monash University, Melbourne, Australia

Abstract—With the growing popularity of social networking, traditional Internet Service Providers (ISPs) and Telecom operators have both started exploring new opportunities to boost their revenue streams. The efforts have facilitated consumers to stay connected to their favorite social networks, be it from an ISP portal or a mobile device. The use of Web 2.0 technologies and converged communication tools has further led to a rise in both user-generated content as well as contextual information (i.e. rich presence) about users – including their current location, availability, interests and moods. In this evolving landscape, social networking players need to innovate for value-centric usage models that increase customer stickiness, along with business models to monetize the social media. To this end, we present R-U-In? – an activity-oriented social networking system for users to collaborate and participate in activities of mutual interest. Activities can be initiated and scheduled on-demand and be as ephemeral as the user interests themselves. R-U-In? leverages contextual modeling and reasoning techniques to manage the entire activity lifecycle in real-time. Initial survey results, based on a prototype implementation of R-U-In?, attest to the promise of real-time activity-oriented social networking – both in terms of an effective collaboration tool for value-oriented social networking users and an enhanced end-user experience.

Index Terms—Social Networks, Converged Communications, Presence, Web 2.0, Semantic Web

I. INTRODUCTION

Recent years have witnessed a surge in social networking as a popular medium for establishing online communications and relationships. Communities like MySpace, Facebook, Orkut, LinkedIn, and Friendster, have rapidly grown from being fashionable hang-outs to portals of the future. They serve as consolidation points for communication in a digital world cluttered with devices, channels and media. Of the many reasons that draw social networking users, the primary one seems to be the ability to communicate with friends, family, and colleagues. A simple yet intuitive notion of a “wall” (in Facebook) has turned out to be the basic feature of social networks. Secondly, social networks provide a unique platform for self-expression. User-generated content (ranging from YouTube videos to Twitter blogs) has given rise to an unprecedented social phenomenon. Finally, social networking allows users to effectively exploit what has been its anchor feature, namely the diverse interests of the users themselves. It is these interests (often expressed as part of a user’s profile) around which new contacts are established, communities are formed, and new opportunities to collaborate are created.

Like with any other technology, what attracts people initially is often not what keep users interested and vested in the long run. This is especially true for one driven by the community itself. Hence, there would be a tipping point where social networking users get tired, lazy or lose the initial excitement and churn out. While it is debatable when this point would be reached, social networking players are already looking at strategic initiatives to sustain and monetize their networks. We believe that this would largely involve an image make-over of social networks from its broad-based usage (e.g. writing on the wall, browsing, viewing photos and “poking” each other) to a more value-centric one that could possibly include search, advertising, commercial transactions, and other productivity-focused behavior.

To provide such value-adds, most agree that it is inevitable for online social networks to open up their services for integration with third parties. Facebook is possibly the first to open up its services, and is being quickly followed by others. Technologies like Web 2.0 [23] have started to play a pivotal role in these developments, in terms of the ability to consume and integrate data from multiple information sources. At the same time, with increasing linkages between mobile and Internet services, Telecom operators are looking at ways to harness the potential of social networking. The efforts are not limited to bringing social sites to a mobile format. Operators have a wealth of content (e.g. user records) associated with their network, along with core network enablers, like converged communications, location, and presence. With evolving architectures like IMS [10] and standards like Parlay-X [19], it looks increasingly feasible to combine network capabilities with Web-based services, and create new revenue streams for operators in a Web 2.0 world.

Consequently, “contextual communications” has emerged as one of the key features of emerging social networks, i.e. where consumers are actively using the end device (mobile,
PDA, or laptop) and a plethora of communication modalities (SMS, Google Talk™, IBM Sametime™) – updating their status, availability, mood and even current interests to drive enhanced levels of communication. *Rich presence*, in this context, is thus not limited to the location and/or availability, but captures a plurality of attributes that directly extend a user’s personality. Using technologies like Web 2.0 and converged communications, one can then envision this rich presence being catered to social networking communities – thereby breeding a new genre of real–time communications driven social networking. In specific, user interests form the anchor of social networks. Interests spawn “activities” and activities lead to collaboration. Such collaboration (be it a movie outing or a business seminar) not only leads to value–addition, but also opens up untapped revenue streams for the social networking portal (e.g. each time a movie is scheduled, revenue is collected from the theater).

Social networks of today are yet to fully realize and exploit the domain of collaboration. They do capture a lot of contextual information about users ranging from hobbies to location to even “What are you doing now?”–style Facebook feeds. To add to this, there is a lot more contextual information (i.e. rich presence) that can be collected from multiple communication modalities available to users. What is missing, however, is a way to consume all this information and subsequently put to use. To this end, we introduce the paradigm of activity–oriented social networking. An *activity*, shown in Figure 1(a), can be initiated (created) by any user and scheduled with other participants who have mutual interest in the purpose of the activity. The participants could be drawn from existing social contacts as well as new contacts who have similar interests. Further, the interests need not be as limited (and as slow changing) as those expressed via a typical social networking profile. Users can define their own real–time interests, update them as frequently as they wish, and finally get grouped together “on–the–fly” for an activity, that is possibly as short–lived as the interest itself. Finally, once the activity is scheduled, it is managed in real–time based on the communication preferences of the participants (both on– and off–portal). Clearly, social networks of today do not cater to the requirements of such real–time activity–oriented networking. Some of them (e.g. Jaiku) have utilities for live presence updates to buddies (interests, location etc.) leaving the onus of creating and managing activities to the users. Others (e.g. Facebook Events application) offer a bulletin–board style approach for soliciting participation in upcoming events. The approach is again limited to a user’s known contacts. Further, they mostly require a user to be logged on to the portal and do not leverage the potential of social networking integrated with real–time communications.

In this paper, we introduce R-U-In? – a system for activity–oriented social networking that exploits the benefits of rich presence and converged communications, in real–time, leading to effective collaboration and an enhanced user experience. R-U-In? draws upon the strengths of Web 2.0 and converged networks technologies to create an up–to–date contextual information store about participants and their interests, and subsequently applies contextual modeling and reasoning techniques to enable “social search” on this platform. In doing so, R-U-In? is a step in the direction of truly ubiquitous social networking – that does not require a user go to a web site to be social, but one that seamlessly integrates with the “social” context in our personalities, our communications, and our activities.

**II. CHALLENGES**

**Scenario:** Joshua wants to catch the latest James Bond sequel on Friday night after work. Before leaving office, he tries to call/SMS his regular movie partners only to find that they have other plans. Joshua then logs on to the new R-U-In? portal – he simply issues a search (South...
Delhi, Movie, Quantum of Solace, 9PM) and is returned with potential matches (on Google Maps™), who are located in the vicinity, available at that time, and have expressed “similar” interests. Joshua can browse over each match and follow links to their social networking profiles. In particular, he likes the profile of Sana who is not included in his buddy list, but happens to be a “Facebook friend” of his friend. Using the click of a button, Joshua sends an invite to Sana to be delivered to her portal and (an SMS) to her mobile device. Once Sana accepts the invitation, a notification is sent back to Joshua’s preferred communication tool. He can now confirm the event and further use the on–portal communication widgets (e.g. SMS, Click-to-Call, IM) to fix a meeting point with Sana. The two eventually meet up and enjoy the movie. Once back home, Joshua adds Sana to his buddy list for future activities.

“Real–time” activity–oriented social networking, as described in this scenario, brings its own challenges that state–of–the–art systems are not designed to handle. To understand the missing links, it is important to first understand the design requirements imposed by activity–oriented networking. Figure 1(b) captures a summary of these requirements. To begin with, the proposed system needs to be designed around user interests and activities – where each activity is best defined as a collaboration involving a mutually interested group of users. People with common interests can then be grouped together as buddies (friends or contacts in social networking parlance). Having said so, the system must enable ways for a user to easily find new buddies who are interested in such collaboration. This feature of interest–based "social search” (i.e. outside a user’s contact list) is one of the key differentiators of the proposed system.

The search process itself is driven by user interests that possibly match the interest around which the activity has been proposed. In contrast to traditional profiles on social networking sites, interests can be very unique (and diverse) – thereby triggering a long–tail of activities to participate in. In addition, the interests are often short–lived in nature. Joshua, for example, can be interested in a “James Bond movie” on one Friday but a "rock concert” the next Friday. He might be also interested in "tennis” on Saturday morning and "community service in South Delhi” on Sunday. In summary, the interests are dynamic, ephemeral and highly contextual in nature. The context might span beyond spatio–temporal and availability attributes to reflect, moods, expressions, or even opinions (e.g. “horror” movie, “democratic” fund-raiser etc.). We refer to this context as rich presence. Unlike searching based on keywords, any form of matchmaking that is designed must exploit this underlying layer of rich user presence.

Where the interests are transient, activities are bound to be scheduled on–demand and short–lived in nature. Once again, this is in contrast to the traditional long–running social communities and the planned events therein. To support this pool of dynamic activities and engage the users in real–time, system responsiveness becomes a key challenge. Users must be able to connect and communicate their interests whenever and wherever they want. With maturing Web 2.0 technologies and converged communications, there are a plethora of such options to do so – through a mobile device (e.g. SMS, application–based updates like Jaiku, or SIP[20]–based availability updates), through a messaging client (e.g. Skype, Google Talk, or enterprise clients like IBM Sametime), and through social networking portals (e.g. Facebook, Twitter, Digsby). The system thus needs to act as a gateway to collect and aggregate rich presence feeds from these multiple (distributed/federated) sources. In addition to collecting information, the system must be capable of disseminating information to its users in real–time – be it a presence update of a buddy, a new activity that has been created and could be of potential interest, or changes in an activity that the user is scheduled to participate in (e.g. members joining/leaving, change in time etc.). Further, the system needs to be flexible enough to choose a communication modality based on the user’s preferences.

Finally, the system needs to handle privacy policies as mandated by the user, so that any contextual information is shared only with the intended audience and any malicious use of the system is prevented at all costs. Thus, trust and reputation are other factors that must be addressed to make the vision of activity–oriented social networking a reality.

III. DESIGN PRINCIPLES

We next describe the design of R-U-In?, a real–time activity–oriented social networking system, that addresses some of the challenges identified earlier. Figure 2 gives an overview of the proposed system that consists of the following functional layers: (i) an interaction management layer, (ii) a knowledge management layer, and (iii) an activity lifecycle management layer.

We elaborate on each of these functionalities before presenting the detailed architecture.
A. R-U-In? Interaction Management Layer

This layer is responsible for the control and co-ordination of all interactions between the system and its users. Users connecting to R-U-In? are envisioned to use a plurality of communication tools that include mobile applications, instant messaging clients, and social networking portals. The interaction management layer, thus, acts as a gateway for users to (a) feed rich presence updates to the system, (b) search and create activities of interest, and (c) stay up-to-date with real-time changes in scheduled activities. Web 2.0 technologies (like AJAX) and next-generation communications paradigms (like Parlay-X/SIP) enable the interaction layer to seamlessly connect and communicate with end-users, irrespective of their access mechanisms. Further details are presented in section IV. Finally, the interaction management layer draws from principles of context-aware communications [21] – thereby allowing users to define policies (preferences) for the system to interpret and dynamically choose an appropriate communication modality (e.g. “SMS from 9AM–6PM, Google Talk at other times”).

B. R-U-In? Knowledge Management Layer

Traditional social networks typically allow users to solicit participation from known contacts. However, activity-oriented systems work the other way round by finding potential contacts for a given activity. The context (interest) of the activity drives a matchmaking process from a repository of real-time interests. A well-defined context model is therefore proposed to store and retrieve context-data and subsequently reason with it. Further, the knowledge management layer is responsible for contextual modeling and reasoning based on activity interests, and is one the core artifacts of R-U-In?.

**R-U-In? ContextModel:** The ContextModel models tags provided by users as part of the (activity/real-time) interests. We consider each interest to be tagged along three dimensions: (i) location description (ii) category description, and (iii) time description. A user can then specify an interest using one or more combinations of these tags. For example “PVR Priya, Movie, 7PM”, “Mainland China, Afternoon” or “Delhi, NightClub” are different possible ways of expressing interests. Tags can often contain implicit (contextual) information about the dimensions (e.g. “Afternoon” instead of 12PM, or “PVR Priya” instead of Movie Theatre). It is the onus of the ContextModel to interpret the tags along as many dimensions and reason about them. Ontologies are promising to model context information because of their expressiveness and reasoning capabilities. Figure 3 illustrates the R-U-In? ontology-based ContextModel.

Tags are modeled as **concepts** in our ontology. Concepts in the category dimension and temporal dimension are derived from the concept **Interests**. Concepts in the category dimension are hierarchically organized using **subClass** relationship with **Interests**. **Instances** of concepts (e.g. “Quantum of Solace”) are attached to concepts that best describe the instance. **Interests** can be (optionally) related to similar other **Interests** using **relatedTo** symmetric property.

The ContextModel is implemented using OWL Full [17] semantic markup language (refer Fig.4). Concepts in the temporal dimension of this model inherit properties from **TemporalEntity** concept. This concept represents a time interval in a day based on the well-defined OWL-Time semantics [18]. Each **TemporalEntity** concept has a **start** and **EndTime**. R-U-In? allows temporal overlaps between concepts, in order to capture tags that represent time-intervals. Further, we define **temporallyRelevant** as an **objectProperty** belonging to **Interests** whose **Range** points to a **TemporalEntity** concept. This captures the relationship between a concept in the category dimension (“Horror Movie”) with a concept in the temporal dimension (e.g.”LateNight”).

To specify the location in a user interest is equivalent to geo-tagging the interest. By using popular GIS tools (e.g. Google Maps) R-U-In? is capable of finding geo-tagged interests that are in spatial proximity. In addition, R-U-In? uses a the location ontology to capture semantic (contextual) relationships between geo-tagged locations. For example, two spatially “distant” theatres can still serve as locations for movie interests in a neighborhood (e.g. “south Delhi”).

Our model captures this using a **connectedTo** relationship (**SymmetricProperty**) between two **Location** concepts. Each **Location concept** is related to one or few location concepts through a **partOf** relation. This relation helps establish the layering between various locations in an intuitive sense.

**Matchmaking Methodology:** This is a two-step process. First, given an activity description of the form <location, category, time>, the ContextModel is consulted (reasoning) to return semantically related set of tags. These tags are then compared (keyword match) with the real-time user interests stored locally. Storing the user interests separately keeps the ContextModel tractable, while employing scalable data management techniques for handling high volume updates to user interests, availability etc.
R-U-In? uses Pellet [6], an open source OWL Reasoner, and supports JAVA–based APIs to reason and return matching tags along the three dimensions. Further, it supports APIs to reason about cross–dimension relations (e.g. “Horror Movies” happens at “Night”). Several in–built OWL semantics (owl:disjointWith, owl:equivalentClass, owl:SymmetricProperty etc) are used in handling different dimensions of the ContextModel. Finally, it supports parameter tuning to control the scope (number) of matches returned. Based on the search criteria specified by a user, an activity like “The Happening” can then be matched to an interest in (exactly) “The Happening” or “Horror Movie” or (loosely) any “Night” activity.

As with other semantic systems, the knowledge management layer needs to be bootstrapped with concepts, their relations and instances. For the initial prototype, this bootstrapping is done manually – while the time (calendar) and location (country/region–wide GIS) dimension have standard reference points, the category ontology is populated to capture common interests and their instances. Relations between interests or across dimensions are then included on a common sense basis. In future, we envision the knowledge base to also support automatic population based on user–tags, thereby creating a rich and diverse set of interests (activities).

C. R-U-In? Lifecycle Management Layer

The (activity) lifecycle management layer controls and co–ordinates the functionalities that are involved in creating and scheduling activities “on–the–fly”, tracking changes in activities (e.g. members joining/leaving, updates in time or location), and finally deleting activities that have concluded from the system. In summary, this layer is responsible for managing the end–to–end lifecycle (in Fig. 5) of the activity corpus within R-U-In?.

To do so, the lifecycle management layer maintains a repository of information about both users (i.e. profiles, preferences, and prescribed communication channels) and activities (i.e. real–time attributes pertaining to each activity).

In particular, for each activity that is “alive” (where an activity is alive from the time of creation to completion), the repository uses appropriate data–structures to store (i) the time (and/or duration) of the activity, (ii) location of the activity (landmark, region, and/or GIS co-ordinates, etc.), (iii) creator of the activity, (iv) list of participants, (v) purpose/interest of the activity as described by the creator, and (vi) rules/policies specific to the activity itself. As explained earlier, many of these attributes serve as tags that aid in the contextual search performed by the knowledge management layer.

In addition to the content, this layer has the requisite control logic to perform actions at different steps of an activity lifecycle. For example, on receiving a search request from a user, it needs to consult the knowledge management later for finding interested participants and subsequently confirm their participation. Similarly, when an activity attribute changes (in real–time), it needs to co-ordinate with the interaction management layer to push live updates to the participants. In the next section, we provide a detailed architecture that puts these functionalities of together in a holistic manner.

IV. Architecture

In the following sections, we present the architecture and implementation details of the R-U-In? system. Figure 6 shows the components of the system, that have been carefully designed on the basis of principles described earlier. The components belong to one of the following three categories: (i) the Rich Presence Gateway that serves as the entry point for rich presence (e.g. real–time interests) from users, (ii) the Converged Communication Platform that provides multiple communication channels for the user to interact with the system (e.g. create activities), and (iii) R-U-In? Core that manages the content, context, and functionalities to create and co–ordinate interest–based activities in real–time. We next elaborate on each component.
A. R-U-In? Controller

The Controller is the heart of the system and encapsulates the control logic for managing real-time interests and activities, i.e., the R-U-In? activity lifecycle. We follow an event-based approach where the controller logic encapsulates a set of “events” and a set of “actions” that need to be executed for each event. Some of these actions require coordination with other system components.

Events are generated by way of interactions between users and the system. An interaction maybe session-oriented (e.g., a user creating an activity, sending out invites and then confirming the event), or may involve a simple update – a user-related update (e.g., change in real-time interest) or an activity-related update (e.g., change in location, time, or participants). In either case, the controller identifies the nature of the event and invokes the actions that need to be performed. For example, to create a newly requested activity, it consults the matchmaking engine to fetch a list of potential participants. This list is presented to the user ¹ and once some participants are selected, the controller sends out invites using the interaction manager (described later); forwards responses back to the user, and finally confirms/deletes the proposed activity. Similarly when a user updates his/her real-time interest the controller needs to update the contextual information repository. Other actions for updating the repository whenever a user’s location/availability changes, sending updates to the buddy list whenever a user’s presence changes, sending updates to participants whenever there are changes in an activity (like start time), registering new users, and registering new activities are encapsulated within the Controller logic.

Finally, the Controller also uses a policy repository that can capture a range of policies related to users (e.g., preferred communication modes), their real-time interests (e.g., privacy rules), or their activities (e.g., rules pertaining to the group).

B. R-U-In? User Interaction Manager (UIM)

The User Interaction Manager (UIM) provides a rich set of interfaces for the R-U-In? system to communicate with its user population. Communications between a user and the system is bi-directional in nature. While a user’s real-time interest is an example of a user-to-system communication, there might be other (rich) presence attributes that can also be communicated. Some of these are directly communicated by the end-user (i.e., interests populated through SMS, IM, social networking portals), while others can be collected (inferred) from the underlying infrastructure (i.e., location information from a Telco’s location server, availability information from a SIP-based enterprise presence server). Additionally, the UIM also supports system-to-user communications – an example being an activity invite sent to the list of potential participants using their preferred communication modalities (SMS, IM, or Web portal).

¹Presentation details, i.e., data format for the response, how much meta-data to include, etc. are captured as part of the Controller logic.

Fig. 6. R-U-In? - architecture details

To enable converged communication capabilities, the UIM interfaces with multiple adapters. The Rich Presence Gateway (in Fig. 6) comprises of one set of adapters that collect “presence” feeds from multiple sources. This includes (i) a PARLAY adapter for SMS-based interest updates and location updates from the Telco infrastructure using Parlay-X [19] based Web services, (ii) an Instant Messaging (IM) adapter for IM client-based updates using protocols like XMPP [24], (iii) an HTTP/SIP adapter for interest updates from any HTTP or converged (HTTP+SIP) application using technologies like Web 2.0 (AJAX) and SIP, and (iv) a Social adapter for updates from third-party social networking portals using JAVA/Web Services APIs. The Converged Communications Platform refers to the second set of adapters that allow the users to interact with the R-U-In? core – to create activities, receive activity invites and other notifications, search for activities etc. In reality, we can re-use the same set of adapters from the Presence Gateway.

Updates received through the Presence Gateway are forwarded to the Aggregator that employs customized presence aggregation rules and store them within a Context Store (described later). Real-time updates might have conflicts. For example, Joshua may be “busy” on Google Talk, while “available” on Yahoo!. Similarly, he might have different interests published using different modes. Context-based conflict resolution is an active area of research and something we plan to look into. For now, we use simple techniques (e.g., OR-ing of availability status, set-union of different interests, highest priority to SMS updates etc.) to handle conflicts.

The UIM also receives system-to-user communications from the Disseminator to be routed via the converged communications platform. In the process, the Disseminator co-ordinates with the R-U-In? Controller to decide preferred modes of communication for the intended recipient(s). To minimize unwarranted updates, the Disseminator can prioritize available modes of communication in descending
order of disturbance factor – e.g., critical updates are sent immediately via SMS, while others delivered when a user logs on to a portal. Policies to define critical activities (e.g. those marked by the creator or scheduled in the near future) can be stored in the Context Store (described later).

C. R-U-In? Context Store

The Context Store manages real–time contextual information about users and exploits this knowledge store to “match” activities to users who would be potentially interested in collaborating. The matchmaking engine (or Matchmaker) can thus be considered as the brain of the R-U-In? system, containing the necessary intelligence required to search for and return users whose interests are relevant to a proposed activity. As discussed in Section III, we apply semantic principles based on contextual modeling and reasoning to perform efficient matchmaking.

Apart from the Matchmaker, the Context Store maintains a repository that contains real–time information (i.e. rich presence) of the users. The information includes their real–time interests, as well as other attributes like location, availability, moods etc. As mentioned earlier, the real–time information is fed to the repository by the Aggregator – which receives these feeds (via UIM adapters) from one or more communication channels located in the consumer, enterprise, or (Telecom) operator network. The Matchmaker consults this repository to construct the list of matching users. The Controller invokes the Matchmaker whenever a new activity needs to be created. In addition, it is also be used for “suggesting” already confirmed activities to users (e.g. each time a user updates her real–time interests).

D. R-U-In? Content Store

Apart from contextual information, R-U-In? needs to maintain information about users of the system, activities that are created in and subsequently removed (after completion), and policies pertaining to both users and activities. The User Profile and Policy Repository records information supplied by a user, possibly at time of subscription. The profile information contains demographics data, as well as data that helps describe the personality of the user (analogous to profiles maintained by traditional social networks). In fact, the R-U-In? profile contains links to other social networks of the user. It also maintains user–specific policies that can be specified at subscription time (e.g. preferred communication modes, privacy filters etc.) and modified later. Interfaces to store and retrieve profile and policies data are provided.

The Content Store also maintains information about activities in an Activity Groups Repository. Information like creator and members of an activity, attributes of the activity (time, location, description etc.) are maintained as part of this repository. It also provides interfaces for creation, deletion and modifications to activities, as well as retrieving activity–related information. It might also be possible to store/retrieve policies that pertain to the activity group as a whole (e.g. communication modalities specific to the group) as part of the repository. Going forward, one could envision populating this store with photos, videos, blogs etc. of activities.

V. Implementation

We have completed a prototype implementation of an activity–oriented social networking system. R-U-In? is deployed as a JAVA–based application on IBM Websphere Application Server V6.1. The server application implements the different components and functionalities of its Core (discussed earlier). In particular, the Matchmaker exploits OWL 1.0 ontologies for contextual modeling of real-time interests, and Pellet – an open source OWL DL reasoner.

Adapters: We have implemented a rich Presence gateway that is capable of collecting user interests via SMS, Google Talk client, and Facebook (as shown in Figure 7). The TWSS adapter (implemented on IBM Telecom Web Services Server) uses Parlay–X [19] APIs to receive live SMS–based updates from the messaging gateway of a tier–1 Telecom operator. It is also used to place calls to a user’s mobile device using TWSS Third Party Call Control APIs. The Google Talk adapter uses XMPP–based Smack [11] APIs to communicate with Google Talk server and receive interest updates from a user’s Google Talk client, by way of masquerading R-U-In? as a Google Talk buddy. The Social adapter uses Java APIs to receive interest updates from Facebook by way of masquerading R-U-In? as a user’s Facebook friend. While SMS and Google Talk are used for real–time interest updates, current Facebook APIs are limited to access interests listed as part of user profile. Work is in progress to extend the social adapter to receive real–time interests from sites like Twitter.

Deployment model: As part of the deployment, each adapter is implemented to have dual–interface – where on one side they communicate with the operator/service provider’s network, while on the other they communicate with the IBM Presence Server (PS). Thus, each adapter implements a SIP interface through which they transcode real-time updates into a standardized PFD [22] document and PUBLISH it to the PS. The Aggregator inside R-U-In? Core SUBSCRIBEs to the PS and receives updates as part of SIP NOTIFY messages. This enables a distributed deployment of adapters – connected to the core using a commercial–grade Presence Server scalable enough to support high volume updates.

Web Portal: We have implemented an R-U-In? Web portal using Java Server Pages, JSP, and AJAX technologies. Using the portal, a user can search for activity partners. Potential matches are displayed on Google Maps (Fig. 8(a)). The user can further use the different tabs to Invite Buddies, Manage Invites (to accept an invite in Fig. 8(c)), or Manage Activities (to confirm an activity and add new contact to buddy list in Fig. 8(d)). Further, when a user updates her own interests, the portal suggests a list of confirmed activities that she could request to join (Fig. 8(b)) in real–time.

\[http://www-01.ibm.com/software/webservers/appserv/was/
\[http://www-01.ibm.com/software/pervasive/serviceserver/
\[http://www-01.ibm.com/software/pervasive/presenceserver/\]
Finally, she can employ the on–portal SMS/Call/IM widgets to establish real–time communications with her partner via the corresponding adapters. Work is in progress to create new activities using (interactive) SMS, in cases where the user is not logged on to the portal.

VI. EVALUATION

The R-U-In? prototype was demonstrated as an activity–oriented social networking solution to a group of users and a survey was conducted to collect and interpret user feedback. We next present the results from this survey.

Survey Process: There were two main objectives that guided the user survey. First, we wanted to gain an insight into the present day requirements of users, i.e. how they would like to go plan activities and find interested partners for them, and how they usually meet these requirements in absence of the proposed system. Secondly, we wanted to understand the overall receptivity of users towards a solution like R-U-In? and identify potential enhancements to the solution.

Twenty people were considered for the survey. The subjects were in the age group of 21–36 years and had a operational know–how of both social networking and communication technologies. Each subject was given a demonstration of R-U-In? that included key steps of creating a profile, creating a new activity, inviting potential matches, confirming an activity, and finally updating her real–time interests using SMS, Google Talk and/or the portal. The subject also witnessed how messages were sent by the system to his available communication modes, e.g. when one of her buddies updated interests, when another user invited her to an activity etc. Each subject was finally asked to fill up a questionnaire (with 15 questions) regarding the usability and acceptability of R-U-In?.

A. User Profiling

Table I highlights our main observations. To begin with, most of the subjects maintained social networking profile on Facebook, Orkut etc. and spent approx. 30 minutes on them. All of them planned activities (tennis, movie, restaurants etc.) on–the–fly with partners (about 20% of which got cancelled due to lack of quorum). Users mostly preferred a call or Instant Messaging (e.g. Google Talk) to enquire and plan these activities. Fewer prople preferred SMS (52%), Email (20%) etc. Further, people have to manually decide whom to invite (for what activity) and hence typically resorted to known contacts. However, given an option they would definitely like to explore new and interesting contacts. Finally, there was an equal amount of interest evinced by users to participate in activities that have been initiated by others (known contacts or otherwise).

Key Insights: There are ample opportunities for tools like R-U-In? to help users plan and manage activities. Further, such tools should include ways to express real–time interests, availability etc, using multiple communication modes, thus enabling users to stay connected at all times. Rich presence and converged communications would play a pivotal role here. Finally, while existing contacts are likely to seed activities, the paradigm of “social search” can help users to include new contacts and grow their activity groups, thereby creating a community effect.

B. User Evaluation of R-U-In?

Table II highlights the key observations, in terms of the Yes, No, Maybe answers provided in the questionnaire. Almost all subjects realized the potential of R-U-In? for finding interested partners and scheduling real–time activities. Of the 30% people who were “unsure”, some showed a preference to having limited close contacts. Others opined that it would depend on the usage experience over a period of time, citing issues like search relevance and privacy to be critical in getting them hooked.

A significant 60% of user found R-U-In? to be more effective than the manual modes (like call or SMS), primarily due to its open–ended search capabilities. Interestingly, of the 35% unsure subjects, quite a few mentioned that the scope (e.g. does it cover all my known contacts?) of search results would be crucial. With respect to the “real–time” features for interest updates and communications, an overwhelming 95% agreed with the uniqueness of R-U-In?.
fraction (75%) of the subjects found it to be more effective than the calendaring solutions. Those who were unsure felt the need for well-defined privacy filters (for example, to avoid spamming from remote contacts).

About 70% of subjects mentioned R-U-In? would be useful in scheduling time-critical activities. Un-solicited invitations were once again issue. While we have considered preferences (and disturbance factors) as part of our design, we believe that it is possible to judge the utility of these features only after extended use of the system. Finally, a few subjects argued that the matchmaking process should consider social graphs (friend-of-friend linkages) to direct the search to activity partners who can be “better trusted”.

**Key Insights:** The quality (relevance) of search results that matches activities to potentially interested users was generally identified as the anchor feature of the proposed solution. Thus, advanced matchmaking based on semantic principles and rich user presence is almost mandatory for the R-U-In? system. Further, search results should span across known contacts as well as remote ones who can be added to one’s interest list – while not compromising on trust and privacy issues at any time. Moreover, the use of real-time communications via multiple modalities go a long way in helping users stay engaged to the system and creating value-oriented collaboration opportunities.

**VII. Related Work**

There exists a significant body of work in the domain of community discovery and community formation in social networks ([3], [13], [8]). While some of them study how user join or migrate to new communities, others focus on methods to analyze self-configuring communities. Recently, techniques have also been proposed to extract communities from mobile communications data ([8], [7]). We believe that R-U-In? brings with possibly new trends in terms of forming real-time communities (groups) that are ephemeral and highly contextual in nature. Thus, the analysis of these activity-oriented social networks can unveil a whole new set of results.

Collaboration tools and context-based communication technologies ([21], [16]) have similarly been studied earlier. There are also a number of initiatives that address peer-to-peer (P2P) collaborations ([9], [12], [15]) that address P2P collaborations. Similar calendar-based and/or enterprise networking tools are available for collaboration. However, these tools typically work on static profiles and fail to exploit the rich user presence interest-based collaborations in real-time.

Recently, a number of commercial solutions (e.g. Dodgeball, Jaiku, and Plazes) and research initiatives like BusinessFinder [5] have started to integrate Telecom capabilities (like messaging and location) to foster networking between social and business connections. Once again, these solutions are at best piecemeal, since they fail to support the capabilities that are required to control and co-ordinate the entire lifecycle of real-time activities. R-U-In?, on the other hand, offers a holistic solution to users for managing interests and activities as part of their social and professional lives.

Finally, R-U-In? uses OWL concepts to model real-time interests across multiple dimensions. In this respect, there exists work on “place” ontologies ([2], [14]). In addition, [1] talks about limitations of OWL to support spatial reasoning and proposes a combination of spatial database store and ontology-based reasoning to better represent geographic information and resources. R-U-In? currently does not address these issues. Rather, it focuses on modeling geo-tagged locations that are logically related.

**VIII. Conclusions**

Enabling social networking users to make new contacts and participate in interesting activities in real-time, holds great promise for new revenue streams for stake-holders in the converged communications and collaboration domains. It also adds a live dimension to the current social networking experience. In this paper we analyze the challenges imposed by activity-oriented social networking domain and present R-U-In? – a one-stop shop for managing real-time interests and activities. R-U-In? can be seamlessly integrated with popular social networking portals of today. Doing so could pave the way for an ecosystem of value-added applications exploiting rich user presence.

In future, we plan to extend and enhance R-U-In? in different ways – incorporating social graphs in the search process, computing search relevance using appropriate metrics, and incorporating trust and privacy issues are some of them. Capturing rich presence has several implications w.r.t. federated/distributed presence architectures and their scalability. Our ongoing work focuses on some of these. Finally, we plan to perform a small-to-medium scale deployment of R-U-In? in a Telecom operator’s domain.

**References**

(a) Create activity.

(b) Update interests (with real-time suggestions).

(c) Manage new invites.

(d) Manage activities.

Fig. 8. R-U-In? - prototype features


[17] http://www.w3.org/TR/owl-features/

[18] http://www.w3.org/TR/owl-time/


