Profiling and matchmaking strategies in support of opportunistic collaboration

Adriana Vivacqua, Melfry Moreno & Jano de Souza

COPPE/UFRJ – Graduate School of Engineering
Institute of Mathematics
Federal University of Rio de Janeiro
PO Box 68511, Zip Code 21941-972, Rio de Janeiro, RJ, Brazil
{avivacqua, melfry, jano}@cos.ufrj.br

Abstract. With the recent advances in communications technologies and decentralization of work practices, there has been an increase in distributed, remote, computerized work environments. In most systems, individuals work from their personal computer terminals, unaware of their peers. With the change from a physical to a virtual environment, opportunities for collaboration often go unnoticed. In this paper, we focus on how to bring unplanned collaboration about. We present an agent framework to encourage and support unplanned cooperation between people. Agents build user profiles through analysis of their documents and work environment and match them according to their interests, activities and opportunities for collaboration. By matching users' work contexts, needs and resources, we expect to uncover opportunities for collaboration that might otherwise go unnoticed. Resource sharing is facilitated in the hopes of stimulating collaboration between users.

Introduction

With recent advances in communications technologies and the widespread adoption of computers by organizations and individuals, new work practices have emerged. It has become more common to encounter individuals working at their computers and remotely collaborating with others. A tendency towards the decentralization of work has also gained strength, as teams come together temporarily to work on projects. As more organizations adopt cooperative work tools, individuals are led to the establishment of remote collaborations and working together in virtual environments.

In these environments, certain opportunities for interaction are lost: informal hallway conversations and impromptu suggestions that may influence one's line of thought or work are no longer present. At the computer, a person's environmental awareness is seriously limited, with the absence of visual, aural and environmental information constituting a major setback. Not only that, but computer-mediated interactions are inevitably poorer that face-to-face interactions. It's not as easy to get to know and trust someone in the virtual world, or to casually bump into someone you know and might be able to collaborate with.

Instant messaging tools have started changing that somewhat: they provide a means for people to be aware of others they know who are online at the moment and a quick way to contact them if necessary. However, that still requires that users establish the need for communication and actively decide to initiate contact. We believe more can be done to jump start collaboration. Many opportunities for cooperation are lost due to the lack of awareness that they even exist. Individuals don't know of others skills, interests, availability or willingness to participate on a project. User profiling, competence, interest and expertise management and context awareness, are techniques we employ to assist in establishing cooperation opportunities and induce cooperative work. By making users aware of each other, they can better leverage each other's skills, competencies and available time.

Furthermore, individuals very often log on to messaging systems in "invisible mode", so that no one will know they are there or try to contact them. That is an attempt to reduce unwanted conversations, which tend to start when one is caught online (in "available mode"). That points to the fact that messages can be disruptive in a work environment: to a large extent, information about the activities of others is irrelevant in the current working context and only hinders work [37]. This indicates a need for careful control of information flow, to minimize disruption.

Just-in-time information delivery is the study of how to provide information when it is needed, to whoever needs it and in such as way as to not disrupt the individual's work. The "what, when, how and who" questions have grown in importance along with the volume of information available. To address these issues, we employ profiling and matchmaking techniques to filter down the information to be provided and determine the moment and recipients of the information.

In this paper we present Cumbia, an agent-based framework to support awareness and discovery of potential collaboration opportunities. We introduce some background and related work in section 2, move on to describe the CUMBIA framework in section 3, and wrap up in section 4.

Background and Related Work

Computer Supported Cooperative Work, or CSCW for short, is a multidisciplinary research area that focuses on effective methods for sharing information and coordinating activities. CSCW systems are often categorized according to the time/location matrix, as found in [18] (synchronous/asynchronous vs. centralized/distributed). These may be redefined and reorganized to take into account different kinds of cooperative work and the complexity of the processes they involve that need to be supported [6]:

- Ad-hoc cooperative work: brainstorming, cooperative learning, informal meetings, design work, etc. Process modeling support is implemented through awareness triggers.
- Predefined/strict workflow: office automation style systems, represented by simple document/process flow. Examples of such systems are Lotus Notes [32], Active Mail [16] and MAFIA [28].
- Coordinated workflow: as found in traditional centralized software maintenance work consisting of checkout, data processing, check-in,

merging, etc. There are several prototypes for systems that support coordinated workflow: EPOS [8], MARVEL [3] and APEL [9].

- Cooperative workflow: decentralized software development and maintenance work conducted in distributed organizations or across organizations. Here the shared workspace and the cooperation planning are the main extra factors from the process point of view. Example of a system supporting distributed organizations and processes is Oz [3].

We are currently looking at supporting ad-hoc cooperative work, such as what happens in unstructured or loosely structured work environments. In these cases, work groups and teams are highly reconfigurable and are not necessarily predefined from the start of the project. The academic environment is one such example: research teams may be engaged in different lines of work and specialists may join the groups and contribute at different points. They may work as a temporary addition to the group (with a the objective of solving a particular problem, for instance) or they may become permanently involved with the project as a whole.

Awareness Systems

Awareness has received a lot of attention among CSCW researchers in the past few years. Researchers have started to realize the importance of being aware of collaborators in a group work environment. Initially, the focus was on providing video and audio to support cooperation and awareness, but other tools and methods have appeared since.

Several works deal with video interfaces and the use of video to support personal awareness and informal interactions. For instance, CRUISER [34] is a virtual environment that uses audio and video channels to support the dynamic processes of informal social interaction (social browsing). The interface provides virtual hallways where the user can wander at will. VideoWindow [15] is a teleconferencing system that connects two coffee lounges in different (physically separated) offices. This system investigates the interactions (mostly conducted through a large screen) of physically separated people having breaks in the coffee lounges. Portholes [10] and Polyscope [5] are media spaces to support shared awareness that can lead to informal interactions. This system sends office images to users in order to let them know who is busy and what others are doing.

Some proposals involve motivation, incentives and support for cooperation, such as Pinheiro et al. They propose a framework to provide past event awareness, where users are informed of past occurrences, results and work history of each other (which includes evolution of shared data, members' actions and decisions, etc.), so as to better collaborate in the present [33]. Prospect awareness systems that allow individuals to envision the potential benefits of collaboration have been proposed, in an attempt to motivate collaboration [19].

Other research focuses on document- or task-based awareness and on providing information to users about who is working on the same document or performing similar tasks at a given moment [20, 29, 30, 31]. Piazza [20], for example, provides awareness information about others who are working on similar tasks when using their computers, exposing an opportunity for interaction or cooperation. It supports intentional contacts and planned meetings as well. The PIÑAS [30, 31] platform provides potential and actual collaboration spaces, as well as services tailored to support collaborative writing on the Web. These are clear attempts at matching individuals at the moment they share a work context. It is important to take the current context into account, as any cooperation will most likely happen within that context. Many recent papers address awareness in mobile computing environments, where location awareness is a central issue for collaboration [2, 14, 26].

The most basic form of awareness, personal awareness, is currently provided by messenger systems (such as Yahoo Messenger, MSN Messenger, AOL Instant Messenger, etc.). In these systems, the user specifies a contact list the system will monitor. The system displays the contacts' availability and status, and the user's status to his or her contacts. A more specialized collaborative tool, GROOVE [17] introduces concept of "shared spaces" to increase the scope of personal awareness. In GROOVE'S shared spaces, users can be aware of what others in that space are doing and on what spaces' objects they are working.

The first step towards successful collaboration is becoming aware of the opportunity to collaborate. We therefore focus on potential collaboration awareness, and provide users' with information on opportunities for collaboration [31] given their current work contexts.

Unplanned Interactions

A useful classification of the different types of interaction found in work environments is presented by Kraut [27]:

- Scheduled: conversations previously scheduled or arranged;
- Intended: the initiator sets out specifically to visit another party;
- Opportunistic: the initiator had planned to talk to other participants at some point and took advantage of a chance encounter to do so;
- Spontaneous: a spontaneous interaction in which the initiator had not planned to talk with other participants.

Kraut also points out that the majority of conversations are informal in nature and that these are usually short and build upon previous discussions. Conversations occur because one person happens to be close to another at a time when one wants to ask for or provide information. Studies show that these types of informal interactions play a central role in helping workers learn, understand, adapt and apply formal procedures and processes [21]. Few systems have focused on support for opportunistic and spontaneous interactions.

According to Esborjörnsson and Östergren [14], spontaneous interactions are the actions that take place when human and/or computational participants coincide temporarily at a location and interoperate to satisfy immediate needs. A similar viewpoint is adopted in [2], where co-location is central to spontaneous collaboration. Both works deal with mobile computing environments, but provide useful insight for the implementation of virtual work environments, because they identify important factors for the establishment of interactions. Esborjörnsson and Östergren also point out that users are usually involved in several simultaneous activities, which means that great care must be taken when deciding on the composition of information [14].

Information about knowledge, physical and cognitive skills, distance and psychosocial characteristics like trust and attitudes are important to the establishment of a successful collaboration [2]. According to Aldunate, Nussbaum and González, similarity in activity preferences, basic values, interests, hobbies, culture, common history and trust on the other person are some of the most important predictors of successful contact. Individuals possess mental models of themselves and of others and the closer the models, the more likely they will be to have successful interactions with each other.

Matsuura et al. [29] introduce the concept of virtual proximity, which is defined as situations in which users access the same data or users invoke the same application in the virtual environment. We take a similar approach, using an individual's current work context (what one is currently working on) to inform the search for others with whom it might be interesting to collaborate with.

Agent-based Systems

Intelligent agents are entities that perceive its environment through sensors and act upon it [35]. Agent-oriented techniques are being increasingly used in a range of telecommunication, commercial, and industrial applications, as developers and designers realize its potential [23]. Agents are especially suited to the construction of complex, peer-to-peer systems, because they are lightweight and permit parallelization and easy reconfiguration of the system.

It is currently believed that Multi-Agent Systems (MAS) are a better way to model and support distributed, open-ended systems and environments. A MAS is a loosely-coupled network of problem solvers (agents) that work together to solve a given problem [39]. The main advantages of MAS are:

- Decentralization: ability to break down a complex system into a set of decentralized, cooperative subsystems. Many types of organizations are inherently distributed.
- Reuse of previous components/subsystems: building new and possibly larger systems by interconnection and interoperation of existing (sub) systems, even though they may be highly heterogeneous.
- Cooperative work support: ability to better model and support the spectrum of interactions inherent to cooperative work, since software agents can act as interactive, autonomous representatives of humans.
- Flexibility: being able to cope with the characteristic features of a distributed environment such as CSCW, CSCL (Computer Supported Cooperative Learning) and CSE (Cooperative Software Engineering).

CSCW systems are complex distributed systems and there are many good arguments for the application of an agent-oriented approach for software engineering to deal with this class of systems [23] (for instance, agent-oriented decomposition to handle problem space magnitude and agent-oriented philosophy for modeling and managing organizational relationship). Agents have been used in groupware for a long time due to their social abilities [4]. A recent survey of the application of agents in groupware and CSCW can be found in [11, 38]. NEEM [12], Personal Assistant [13] and COLLABORATOR [4] are some examples of agent approaches used in developing collaborative tools. AwServer, CScheduler and E_Places are good examples of agent-based awareness work [1].

The CUMBIA Framework

We have created an agent-supported peer-to-peer architecture where each user has a cluster of agents to assist with knowledge management and collaboration tasks. Agents are in charge of identifying potential cooperation situations and trying to making these come to fruition by providing relevant information in a timely manner. The CUMBIA framework is detailed in the following sections.

Agent Architecture

Each user has its own "agency" (a group of agent service teams) to assist with knowledge management and cooperation tasks. There are four agent service teams that interact to perform specific tasks: User Interface Services, Collaboration Services, Awareness & Matchmaking Services and Knowledge Management Services, as shown in Figure 1. Agent service teams and main functionalities are:

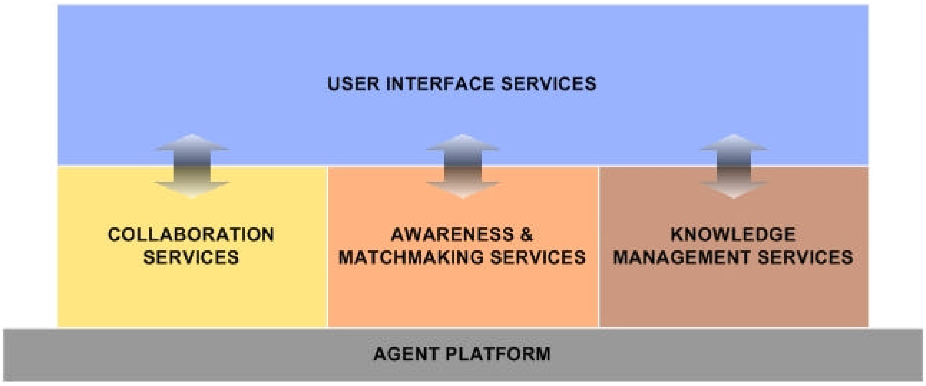

Fig. 1. Agent Architecture

- User Interface Services: information display and allowing the user to specify parameters and information to the other agent teams.
- Collaboration Services: allow for the easy and quick establishment of contact when the possibility for collaboration arises and provide tools for cooperation (forums, messaging, etc.)
- Awareness and Matchmaking Services: search for other users with whom it might be interesting to establish contact, contact other agents for their users' profiles and work contexts, compare user profiles to current context and work environment.

- Knowledge Management Services: manage user's personal data, build initial profiles based upon this data and keep track of document usage, searches, ongoing collaborations and current research.

We created the workspace concept (similar to the multiple desktops found on Linux) to help determine a user's work context. A workspace is a user created work environment: a user can save all the relevant documents, applications, contacts, links, etc. in a workspace when working on a given topic. At any given moment, a user will be working in one of his or her workspaces. A user can easily switch between workspaces, changing work contexts.

In reality, a workspace is only a collection of links, not a representation of physical storage, but a way to determine what resources belong together: documents and links may be saved anywhere on the user's hard disk and be associated with one or more workspaces. When a user switches to a different context, those documents will be the ones in the most frequent list and the bookmarks saved in that workspace will be the ones first displayed (the user can always have the system "show all" in case he/she wants to retrieve an item from other workspaces). This effectively creates context-sensitive bookmarks, documents and work environment. Its is important to note that only one workspace can be active at any given time, although users can allocate items to different workspaces by dropping them into the workspace icons.

Agent service teams and their functionalities are discussed in detail next.

Knowledge Management Services

The knowledge management agent service team provides profiling functionality to the system. Basic units in the profile are projects and interests, which are interrelated. Part of the profile information has to be explicitly provided (mostly links or interrelations between items), and another part is automatically inferred. Users always have the last say on their profiles, being able to correct the information and determine which information can be made public and which is to remain private. In our environment, we consider that a person is always working within a workspace. Our basic context units are workspace definitions. These contain projects, documents, contacts, etc.

Users' projects are related to documents, people, collaborations and research, but are inherent to each user. So, a cooperative project almost certainly will have two or more different project definitions associated with it, one for each team member. This is in accordance with the fact that individuals have personal views of reality and organize their work accordingly. It is useful, of course, to keep track of the correspondence between individual projects that represent the same group endeavor (this entails keeping track of the different aliases a project might have). Projects are usually related to only one workspace, although it is conceivable that a user may be working on two projects at the same time, in one workspace or may have divided a project into more than one workspace.

Profiling Agents keep track of the following information:

- Contact Information: information necessary for another person, to contact the user: Name, Title, Email, Phone, etc.

- Areas of Interest: general areas in which the user has some interest. These may be automatically or manually setup, and will be ranked by interest and activity level. Areas of interest may be related to projects, people, documents and histories.
- Projects: projects the user is involved with. The user creates these project definitions to help with information organization. Projects are classified according to their activity status: Past (project is finished), Present (project is being worked on at the moment) and Future (these are future projects the user intends to work on at some point). Projects may have deadlines associated with them, which can help prioritize agents' work. All other information may be related or not to projects. Projects may be related to areas of interest, people, documents and histories. Project information is manually provided.
- People: a user's contact list, classified into different categories, such as personal or work contacts, previous, current or potential collaborators, researchers, etc. Some information is inferred from email FROM, TO and CC lists. Contacts are linked to projects in the context of collaborations in progress and are linked to areas of interest when the users have similar interests.
- History: agents track pages the user accesses when navigating the Internet. Histories may also be related to projects and interests. Work activity is also logged to create project and collaboration histories that may inform future interactions.

Rating mechanisms will be put in place so that users can rate other users or resources (sites or documents) according to the relevance to the work or project in progress or to an area of interest, providing more information the system can use.

Batch profiling. When first setting up the environment, agents build initial profiles by scanning all available information as provided by the user. The first step is the processing of textual documents, then web histories and contact networks. The following processes are undertaken:

- Processing textual documents: documents are processed using a text-processing algorithm. Keywords are extracted from each document and relations to the documents are established. Initially, we use TFIDF [36], a well-known keyword extraction algorithm, but we plan on testing other algorithms in the near future.
- Processing bookmarked websites: websites from the user's bookmark lists are visited and processed for keywords as well. Relations are built between the links and the keywords.
- Extraction of web histories: websites visited are extracted from the user's web history and searches are separated from site visits. Multiple sites visits are counted and sites are revisited and processed as above. Keywords are extracted from the searches.
- Extraction of contact networks: email archives are processed, with keywords being extracted from the bodies and contacts from the TO,

FROM and CC fields. Contact lists from email and messenger applications are also processed and added to the "master contact list".

Items not classified as belonging to any one workspace are left in a generic workspace, from which they can be retrieved later. This workspace contains documents, links and contacts that haven't yet been assigned their workspaces. Periodically, agents run keyword matches on the generic workspace items and suggest workspace assignments for the unassigned documents. These may or may not be accepted by the user.

Real-time profiling. When the user is at work, his or her behaviors can be observed to furnish some extra information. As the user works, agents observe his or her behavior and process documents in the same manner as when they are batch processed. The following additional information is extracted from the users' behavior:

- Time spent on documents: total time spent on a document is measured
- Access frequency: the system keeps track of how many times the document has been accessed, and when the last access was.
- Access type: the system logs what type of access was made to each document. Possible access types are read (reading the document), write (writing some text), forward (reading and sending it to others) and publish (writing and sending it out to be read by others).

This information can be used to assign weights to certain documents (to represent relevance or importance in that workspace) that can help improve the matchmaking process. We are currently verifying how these variables factor into the process. The following inferences can be made:

- Needs: information needs are extracted from search keywords and documents or references downloaded, bookmarked or saved. These will then be used to search other users' workspaces for relevant information that can be exchanged.
- Resource use: resources are links and documents a user has at his or her disposal. These can be easily shared with other users. Frequency and length of access (vs. document length) determine how important that document is in that context (is it something that was read over and over or referred to several times? Was it printed?) Distribution also denotes importance or relevance: if a user thought the document was good enough to send it to others, that should also be noted, and the links to the contacts should be established.
- Knowledge: documents written by a user define his or her knowledge and documents read knowledge the user is in the process of acquiring (here the length and frequency of access measurements help determine how well the document was studied).

All profile information is saved in a Knowledge Base. The system logs message exchanges (text, email, discussion) and these are linked to workspaces and active projects, forming a personal work history. This history will inform future discussions, assist users in establishing common understanding, and allow users to "pick up from where they left off" when engaging in new interactions. In addition, it establishes patterns of cooperation, helping with the identification of which users have been cooperative in the past (to possibly favor cooperating with these in the future) and which ones have consistently avoided interaction with the user (to possibly avoid these in the future).

User profiling has been largely explored before, and several sources of information have been identified that can be used to build user profiles. Email, bookmarks, source code and publications read and written are some of them. In many approaches, documents are analyzed for their text content and keywords are extracted. We take an existing approach and build on that for profile construction. We run a keyword extraction algorithm (Term Frequency Inverse Document Frequency - TFIDF) [36] on the user's documents to generate weighted keyword lists. This method has been extensively used with good results, so we expect it will work well in this case as well. We will be testing different algorithms in the near future.

We then rate each document according to its importance to the user. Document rating is based on number of accesses, length of access, type of access (read, write, print), and distribution (whether it was sent to someone else or not), as described above. Thus, documents in users' profiles are ranked by popularity, and keyword importance is calculated accordingly. Links between documents and projects and workspaces are used to determine keyword lists for projects and workspaces. These keyword lists are then used in matchmaking.

Awareness and Matchmaking Services

Having built user profiles, matches need to be made. Several studies exist in the matchmaking field, especially in relation to recommendation systems. We identify opportunities for collaboration by matching a user's current context (as determined by the workspace) with other users who might have related interests or work.

An opportunity for collaboration is determined by users' contexts: when two users' contexts are similar or related, an opportunity for collaboration might exist. Given the information needs of each user, we look for documents that match those needs in other users' environments. A search, is a clear indicator of a "time of need", therefore the system looks for matches in other users' workspaces whenever a search is performed (the search performed by the user proceeds unencumbered).

The identification of an opportunity for collaboration is a 3-step process:

1. Given the workspace a user is currently working on, look for other users currently in similar workspaces. Look for active workspaces initially. If none are found, look for inactive workspaces as well.
2. Within those found, look for documents in their workspaces that are similar to the document currently being worked on (and that aren't present in the workspace). If searches are being performed, keywords being used for searching can be used to search other users' workspaces. Check to verify if there is a possibility for reciprocity in the exchange.
3. Ask the document owner whether the documents found can be sent and furnish information on the user who will be receiving it. If owner authorizes sending the documents, ask receiver whether he or she wants to

receive them. If no documents are found, inform both users of a potential collaborative opportunity and of similarity in contexts.

This process begins by finding individuals in similar contexts (it is always better not to have the user change contexts) and then finding resources these users should share. Resource exchange is the most direct way of initiating collaboration, since it places few demands on either user. Once the link has been established, users can be directed to collaborate synchronously and exchange thoughts and ideas regarding each other's work.

Matches are made through keyword similarity calculation. Every document has an associated weighted keyword list, as provided by the TFIDF algorithm. These are compared to determine similarity between items and find possible matches.

Matching can be done "online" or "offline": online matching occurs when the user is working, inserted in a context and the system searches for potential collaboration opportunities in real time. Offline matching runs in the background to find users whose workspaces or documents may be related to projects or areas of interest the user is involved in. This is meant to speed up searches: agents independently pre-search the space to build and store simplified models of other users, which are then used to make initial matches and search in more detail for potential collaboration matches.

Collaboration Services

When an opportunity for collaboration comes up, a user is immediately notified. Opportunities are time sensitive, and the user should be informed of the potential for reciprocity (if any) and should be given information on the other user that includes past partnerships and cooperative behavior. Other useful information, especially for unknown users is to try and find a common link between the two individuals.

After the identification of collaboration opportunity, an individual may become an incidental or an active collaborator in another user's projects. Incidental collaborators provide occasional suggestions and occasionally attend meetings. Active collaborators are inserted in the project and have to deal with schedules and deadlines. It's important to know each participant's status and whether any tasks are dependent on him or her. Project management capabilities are useful in assisting with active collaborations.

The initiation of collaboration should be effortless, so as to create as little overhead to the user as possible. Whenever agents detect some information or document one user has might be useful to another user, they automatically offer to send that user that information, asking only for permission from the owner. In this fashion, a user doesn't have to worry about finding adequate documents, histories or appropriate information to be sent to others. The users may choose to engage in longer interaction, by initiating a chat or message exchange.

All standard collaboration support tools (discussion lists, messaging systems, shared whiteboards, file sharing mechanisms and email) are provided in the system. Most of these exist as modular solutions, which can be plugged in. We concentrate on helping the establishment of first contact and the initiation of collaboration rather than developing new tools. We will progressively add tools and services to the system as they become necessary.

User Interface Services

User interface agents perform interface related functions, provide information to the users and request information from them. UI agents mediate requests between agents and users. Our basic interface displays little information, so as not to disrupt the user. However, most information is easily accessible with a mouse click or rollover.

Workspaces are accessed and managed through the Cumbia Personal Toolbar. The Cumbia Personal Toolbar is a toolbar that sits on a user's desktop, where users can visualize their available workspaces and switch between them, view their contact list, view associated keywords, links, searches and document lists. Every workspace has a resource briefcase where documents and links are stored, plus an address book with contacts related to the workspace and a document briefcase for documents being edited. There is a message bar for system and other user's messages and two collaboration indicators that flash when an opportunity for collaboration is found: one for incoming collaborations (another user can add to your work) and another for outgoing collaborations (you can contribute to another user's work). There is also a workspace viewer, so that users can easily switch between workspaces. Keyword lists that describe the workspaces can be viewed by rolling the mouse over the icons. Users can add to or modify the workspace definitions at will. Workspaces also have a history log.

Initial definitions and assignments must be made by hand (the user has to drag and drop documents into their workspaces), but afterwards documents are associated with the current workspace as the user works on them. Searches will be logged as part of the current workspace, as will documents being written or downloaded and pages bookmarked. A user can always associate an item with another workspace by dropping it into that workspace. If the item is already saved in the current workspace, the system asks whether it should be moved or copied to the other one. In future versions, agents will be able to decide which are the appropriate ways of displaying information when they receive it and the proper time to display it, thus addressing the problems of what, how and when (given that who is fixed). Relevant work has been done in [1], which we will use and build upon in our system. For the moment, opportunities are displayed as small flashing icons, much in the fashion of current messenger systems. The user has the choice of whether or not to click on the icons, receiving more information on the potential cooperation.

Application Scenario

There are three basic types of work environments: structured, loosely structured (or semi-structured) and unstructured. In structured work environments, there is a strict plan that should be followed, a meeting agenda or workflow. In loosely structured ones, there is some structure but it is loose and adaptable, only major outline and breakpoints are in place and it is up to the participants to fill in the blanks, creating their own structure as necessary. In unstructured work environments, there is hardly any structure and what is there is highly changeable: who runs what and how, tasks to be accomplished and steps to that end all depend on the moment and on who's present. Everything is configurable by the participants.

Linux is possibly the most obvious example of unstructured cooperation, as is the whole Open Source initiative. Several studies exist regarding the open source community and how and why it works. We do not presume that this model will work in all environments, but we believe it is important to attempt to identify characteristics that can be replicated in other projects. In the Apache project, for instance, mechanisms were put in place to help with software development, discussion and version control and most issues were resolved through voting.

We are implementing a prototype for an academic knowledge management system. Academic work environments are usually very loosely structured and several opportunities for spontaneous collaboration exist. Groups form as common interests appear and individuals come together to work for a period of time (the duration of a project) and disband later (but ties remain, as does the possibility of further collaboration). Cooperation is often externally triggered as, for instance, with the appearance of a new funding opportunity. External funding agencies provide guidelines for projects (among which there is usually the inclusion of a certain number of qualified specialists): in this case it becomes important to identify and bring together a group of interested, qualified people to form groups and write project proposals to take advantage of the opportunities. This seems like an appropriate application domain, since most of the time, students and academics don't mind sharing resources with each other or entering into collaboration.

We chose JADE (Java Agent Development Framework) [22] to develop our agents. JADE is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of tools that supports the debugging and deployment phase. The agent platform can be distributed across machines (which do not need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be changed at run-time by moving agents from one machine to another one, as and when required.

JESS [24] is used for inference making, specifically deciding when and how information will be shown to the user. Jess was originally inspired by the CLIPS [7] expert system shell, but has grown into a complete, distinct, dynamic environment of its own. Using Jess, we can build Java software that has the capacity to "reason" using knowledge supplied in the form of declarative rules. Jess is small, light, and one of the fastest rule engines available.

The system is currently under implementation, and we expect to initiate testing soon.

Conclusion and Further Work

There are six basic functions of informal communication [21]: tracking people, taking or leaving messages, making meeting arrangements, delivering documents, giving or getting help and reporting progress and news. Most of these can be automated by computers in a cooperative work environment, reducing the need for informal communications. However, informal communication is central to the establishment of a community and strengthening of ties between members. Spontaneous, unplanned interactions are much less frequent in computer based cooperative work environments, where each user works at one station than in offices where users are in physical proximity. We should not be looking to reduce informal communication, we should be trying to increase it.

We have presented an agent-based architecture to support and encourage spontaneous interactions in virtual environments. The first step towards this is identifying potential collaboration situations and making the act of collaboration as effortless as possible. CSCW researchers, recognizing the importance of awareness information, have been striving to provide it in their systems. However, there has been little or no focus as to why it is provided. By focusing on the reasons for providing awareness information, we expect to reduce information flow and create effective mechanisms to encourage collaboration. The workspace metaphor is particularly useful because it helps establish a context for work while assisting the user in the organization of his or her resources. Linux users like working with the multiple desktops and miss it when the y move to Windows.

Two basic problems are always associated with the provision of awareness information: privacy violations and user disruption. Users' privacy may be violated by making details of their activities available that should have been kept private. Every piece of information about a user that is made available to others is a potential privacy violation. Besides, users may be disrupted from their work because unneeded information about others distracts them. For an awareness system to be effectively used, users must trust it. They should be able to understand its limits and capabilities and feel confident that they know what information of their actions can be observed [25]. In our system, the user is allowed to determine what information will be made public, becoming available for awareness purposes. Our system also takes the user's context into account: we provide information that is relevant to the user at the moment, so as to not worsen the problem of information overload or disrupt the user's flow of work or line of thought.

Profiles contain a wealth of information so we can test different matching techniques and variables to determine which work best. There is still much work to be done, namely in the areas of context inference and rule building. We are implementing the first prototype and hope to have some initial results soon. We will be testing and improving on matching and profiling methods as the project evolves.

References

1. Alarcón, R. and Fuller, D. Intelligent Awareness in Support of Collaborative Virtual Work Groups. In: Haake, J. M. and Pino, J. A. (Eds.) CRIWG 2002, LNCS 2440, pp. 147 – 167, Springer-Verlag, 2002
2. Aldunate, R. Nussbaum, M. and González, R. An Agent Based Middleware for supporting Spontaneous Collaboration among Co-Located, Mobile and not Necessarily Known People. Workshop on Ad hoc Communications and

Collaboration in Ubiquitous Computing Environments, CSCW 2002, New Orleans, USA, November 2002.
3. Ben-Shaul, I. and Kaiser, G. A paradigm for decentralized process modeling. Kluwer Academic Publishers, Boston/Dordrecht/London, 1995.
4. Bergenti, F., Garijo, M., Poggi, A., Somacher, M. and Velasco J.R. Enhancing Collaborative Work through Agents. VIII Convegno dell'Associazione Italiana per l'Intelligenza Artificiale, 2002.
5. Bourning, A. and Travers, M. Two approaches to casual Interaction on Computer and Video Networks. Proceedings of International Networking Conference, 1991.
6. C. Liu and R. Conradi. Process View of CSCW. In Proc. of ISFST98, Ocon Technology Application, page 12, Bremen, Germany, 15-17 September 1998. InternationalWorkshop on Intelligent Agents in Information and Process Management.
7. CLIPS, http://www.ghg.net/clips/CLIPS.html
8. Conradi, R., Jaccheri, M.L, and Mazzi, C. Design, Use and Implementation of SPELL, a language for Software Process Modeling and Evolution. In Proc. Second European Workshop on Software.
9. Dami, S., Estublier,J. and Amiour, M. APEL: A Graphical Yet Executable Formalism for Process Modeling. In Process Technology edited by E. Nitto and Alfonso Fuggetta, pages 61–96, Politecnico di Milano and CEFRIEL, 1998. Kluwer Academic Publishers.
10. Dourish, P. and Bly, S. Portholes: Supporting Awareness in distributed Work Group. Proceedings CHI, 1992.
11. Ellis, C.A. e Wainer, J. Groupware and Computer Supported Cooperative Work. In Weiss, G. (Ed.) Multiagent, Systems, MIT Press, 1999.
12. Ellis, C.A., Barthelmess, P., Quan, B. e Wainer, J. NEEM: An Agent Based Meeting Augmentation System. Technical Report CU-CS-937-02, University of Colorado at Boulder, Computer Science Department, 2002.
13. Enembreck, F. and Barthès, J. P. Personal Assistant to Improve CSCW. Proceedings of the 7th International CSCWD, Rio de Janeiro, 2002, pp 329 – 335.
14. Esborjörnsson, M. and Östergren, M. Issues of Spontaneous Collaboration and Mobility. Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings, UBICOMP'02, Göteberg, Sweden, 2002
15. Fish, R. S., Kraut R. E. and Chalfonte, B. L. The VideoWindow System in Informal Communications. Proceedings CSCW, 1990.
16. Goldberg, Y., Safran, M., Silverman, W. and Shapiro, E. Active Mail: A Framework for Integrated Groupware Applications. In D. Coleman, editor, Groupware '92, pages 222–224. Morgan Kaufmann Publishers, 1992.
17. GROOVE, www.groove.net.
18. Grudin, J. Computer-Supported Cooperative Work: History and Focus. In IEEE Computer, number 5 in 27, pages 19–26, 1994.
19. Hoffman, M and Herrmann, T. Prospect Awareness – Envisioning the Benefits of Collaborative Work. Available online at: http://iundg.informatik.uni-dortmund.de/iug-home/people/MH/ProspectAwareness/PAhome.html
20. Isaacs, E.A., Tang, J.C. and Morris, T. Piazza: A desktop Environment Supporting Impromtu and Planned Interactions. Proceedings of CSCW'96, Cambridge, MA, 1996
21. Isaacs, E.A., Whittaker, S., Frohlich, D. and O'Connail, B. Informal Communication Re-Examined: New Fuctions for Video in Supporting Opportunistic Encounters. In K. Finn, A. Sellen and S. Wilbur (Eds.), Video Mediated Communication, Mahwah, NJ: Lawrence Erlbaum
22. JADE, http://sharon.cselt.it/projects/jade/

23. Jennings, N.R. An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM, April 2001/Vol. 44, No. 4
24. JESS, http://herzberg.ca.sandia.gov/jess/
25. Kanerva, A., Koskinen, L. and Pitkäniemi, T. Awareness Support in Virtual Collaborative Systems. University of Helsinki, Department of Psychology and Teamware Group. 2000.
26. Kortuen, G., Gellersen, H.W., Billinghurst, M. Mobile Ad Hoc Collaboration. Proceedings of CHI 2002, April 2002, Minneapolis, USA
27. Kraut, R., Fish, R., Root, B., and Chalfonte, B. Informal communication in organizations: Form, function and technology. In S. Oskamp and S. Spacapan (Eds.), People's reactions to technology in factories, offices and aerospace, The Claremont Symposium on Applied Social Psychology, Sage Publications, 1990. pp. 145-199.
28. Lutz, E., Retzow, H.K. and Hoernig, K. MAFIA - An Active Mail-Filter-Agent for an Intelligent Document Processing Support. In S. Gibbs and A.A. Verrijn-Stuart, editors, IFIP, North-Holland, 1990. Elsevier Science Publishers B.V.
29. Matsuura, N., Fujino, G., Okada, K. and Matsushita, Y. An Approach to Encounters and Interaction in a Virtual Environment. Proceedings of the 1993 ACM Conference on Computer Science, Indianapolis, Indiana, United States, 1993.
30. Morán, A. L., Decouchant, D., Favela, J., Martínez-Enríquez, A. M., González-Beltrán, B. and Mendoza, S. PIÑAS: Supporting a Community of Authors on the Web. Proceedings of Fourth International Conference on Distributed Communities on the Web, Sydney, Australia, April 2002.
31. Morán, A. L., Favela, J., Martínez-Enríquez, A. M. and Decouchant, D. Before Getting There: Potential and Actual Collaboration. In: Haake, J. M. and Pino, J. A. (Eds.) CRIWG 2002, LNCS 2440, pp. 147 – 167, Spring-Verlag, 2002
32. Orlikowski, W.J. Learning from Notes: Organizational Issues in Groupware Implementation. In Proceedings of the Conference on Computer – Supported Cooperative Work, CSCW'92, pages 362–369, Toronto, Canada, 1992. The Association for Computer Machinery, ACM Press.
33. Pinheiro, M.K., Lima, J.V. and Borges, M.R.S. A Framework for Awareness Support in Groupware Systems. Proceedings of the 7th International Conference on Computer Supported Cooperative Work in Design – CSCWD'2002, Rio de Janeiro, Brazil, September 2002, pp.13-18
34. Root, R. Design a Multi-Media Vehicle for Social Browsing. Proceedings CSCW, 1988.
35. Russell, S. and Norvig, P. Artificial Intelligence – A Modern Approach. Prentice Hall, Englewood Cliffs, NJ, 1995.
36. Salton, G and McGill, M.J. The SMART and SIRE Experimental Retrieval Systems. In Readings in Information Retrieval. Karen Jones and Peter Willet, eds. Morgan Kaufmann, San Francisco, 1997
37. Sohlenkamp, M. Supporting Group Awareness in Multi-User Environments through Perceptualization. GMD Research Series Report, No 6, 1999. Fachbereich Mathematik-Informatik der Universität – Gesamthochschule, Paderborn, 1998.
38. Wainer, J. and Ellis, C.A. Agents in Groupware Systems. Proceedings CRIWG, Búzios, RJ, Brasil, 1998.
39. Wang, A., Conradi, R., Liu, C. A Multi-Agent Architecture for Cooperative Software Engineering Proceedings of the Third International Conference on Autonomous Agents, 1999