When Cyborgs Meet: Building Communities of Cooperating Wearable Agents

Gerd Kortuem, Jay Schneider, Jim Suruda, Steve Fickas, Zary Segall
University of Oregon, Deschutes Hall
Eugene, OR 97403 USA
{kortuem,jay,jfs,fickas,zs}@cs.uoregon.edu

Abstract

This paper introduces the notion of a Wearable Community *as a group of wearable users who cooperate for their mutual benefit. In such a community, wearable computers act as personal agents on behalf of and in the interest of their 'owners'. These agents are goal-directed and will perform a broad array of tasks for the user, ranging from personal scheduling to task planning. We describe how personal wearable agents can be used to enable goal directed cooperation during physical encounters of people with selfish and conflicting goals, such that cooperation leads to mutually beneficial results. We discuss negotiation protocols, and describe the design and implementation of a wearable agent system, as well as a simulator for large-scale wearable communities.*

1 Introduction

In our modern world, the use of communication technologies like phone, fax and email has become commonplace. Despite this fact, most human interactions still occur when we meet people face-to-face. Frequently, we use such encounters to cooperate with other people and to pursue and advance our own goals. For example, we purchase items from a salesperson, coordinate schedules with co-workers during a meeting, or make travel arrangements with friends at home.

While the idea of using wearable computers for cooperation and collaboration has been around at least as long as wearable computers have become feasible, today's wearable computers are mostly used in isolation, as advanced personal assistant. A prototypical example for this situation is the Remembrance Agent [11], an augmented-memory application that continuously searches for documents with content relevant to the user's current situation.

A number of wearable systems have been designed with the explicit goal to support collaboration of remote and co-located users. Most of them are wearable videoconference systems [1,2,3,4,8,10,12,17], while others aim at establishing personal relationships between users [5]. For example, our own Proem system [9] enables two or more users to exchange personal user profiles in real-time during face-to-face encounters with the goal to identify shared interests.

All these system have one point in common: they assume that users share common goals and want to cooperate for the mutual benefit of all participants. The shared goal is usually defined by the broader context in which people meet or collaborate, for example by the fact that both users are employees of the same company: two technicians who collaborate using a wearable maintenance assistant share the goal to find and repair an equipment failure. In our daily life, however, we are often faced with the necessity to cooperate with people with different or even contradicting goals. For example, when buying a car from a dealer our own goal might be to purchase a car that fits our lifestyle at a price we can afford, while the car salesperson's goal might be to sell a car from the lot that maximizes his commission. This type of goal-directed cooperation between users with differing or even antagonistic goals is not supported in any wearable system.

In this paper we address the question how wearable devices can be used during physical encounters of individuals to support cooperation. In particular and in contrast to most previous research, we are looking at how this technology can be used to enable *goal-directed cooperation during encounters of people with selfish and conflicting goals, such that cooperation leads to mutually beneficial results.*

We believe that wearable computers provide a unique platform to support this kind of cooperation: they are highly personal, proactive, and context-sensitive. Our emphasis in this research is on large-scale communities of wearable users where people interact and cooperate in many ways and for a variety of reasons.

This paper is organized as follows. In the next section, we start by presenting a theoretical framework of cooperation in wearable communities. In Section 3, we give an example of a negotiation protocol and discuss its properties. In Section 4, we describe the design and implementation of a wearable agent system. In Section 5, we present WALID, an interactive simulator for evaluating the behavior of agents in large-scale wearable communities.

2 Wearable Communities

We define a *Wearable Community* as a collection of wearable users who cooperate for their mutual benefit. Our notion of a community is based on three fundamental concepts:

Personal Agents: Each wearable computer acts as a personal agent on behalf of its user.

Physical Encounters: Wearable users encounter each other as they move around in space.

Agent Cooperation: Agents cooperate through negotiation that take place during encounters.

We will now discuss these concepts in more detail.

2.1 Wearable Agents

We call any wearable computer that exhibits autonomous behavior and acts on behalf and in the interest of its user a *wearable agent* (agent for short).

Agents have several important properties:

Goal-oriented: Agents are goal-directed and will perform a broad array of tasks for the user, ranging from personal scheduling and task planning, to providing hardware services in foreign locales.

Opportunistic: Wearable agents are designed to benefit their users and to act in their best interest. We do not require nor do we assume that agents will *a priori* be cooperative, share information, or negotiate for a single global goal. In particular, we do not make any assumptions about agents being "benevolent" [21].

Predictable: Agents are predictable. They act according to rules defined by their users and can be fully controlled and monitored.

Rational: Agents make rational decisions based on the utility of alternative choices.

The more complex the considerations that a wearable computer takes into account in order to advance its user's goals, the more justified it is to consider this computer an agent.

2.2 Encounters

The second fundamental concept of a wearable community is a physical encounter between individuals. We define an *encounter* as a situation where

- two or more individuals are in close physical proximity to each other;
- the wearable computers of these individuals have discovered each other's presence; and
- these computers are able to communicate.

This definition does not say anything about how wearable computers discover each other, or how close they have to be in order to do so. Many different and equally suitable technologies have been used in the past for proximity sensing, including infrared transmitters and near-field radios. Similarly, this definition does not say whether discovery and communication are independent functions or can be combined into one. The latter would be possible with Personal Area Networks [18] or future short-range wireless networks like Bluetooth [22].

Encounters have several important properties:

- Encounters can occur between two, three or more individuals.
- Encounters are reflexive: if user A encounters user B, then user B encounters user A.
- Encounters are situations with a time duration, not momentary events: encounters can be short and last only a few seconds, or they can be long-lasting going on for hours. For example, the encounter between two individuals passing each other in a hallway might last just a few seconds. On the other hand, two or more people in a lengthy meeting encounter each other for the full duration of that meeting.
- Encounters are non-transitive: if user A encounters user B, and user B encounters user C, then user A does not necessarily encounter user C.

2.3 Agent Negotiation

The third and final key concept of wearable communities is that of *agent negotiation*. An encounter between individuals is a chance for an automated interaction of their respective agents in form of negotiation.

Following Rosenschein and Zlotkin [16], negotiations are defined by *protocols* and *strategies*.

Protocols. A negotiation protocol is the set of rules governing the inter-agent communication among agents – offers and counter-offers, threats, promises, concessions, etc. In particular, negotiation protocols deal not with the mechanism of communication but with its content. A protocol specifies the kinds of deals agents can make, as well as sequences of offers and counter-offers that are allowed.

Strategies. A strategy is the way an agent behaves in an interaction. The protocol specifies the rules of the interaction, but the exact deals that an agent proposes are the result of a strategy that his designer has set. Strategies may involve the relaxation of initial goals, concessions, or even lies.

3 Negotiation Protocols and Strategies

In the preceding section we have introduced our idea of wearable agents and wearable communities in theory. We now discuss concrete negotiation protocols and strategies for wearable communities.

3.1 An Example: Delivering Packages

It may be useful here to look at an example. Suppose that two drivers working for independent delivery services meet each other at several stops along their routes, because they have to deliver packages to essentially the same destinations. Instead of following each other around, they realize that they could trade packages between them in a way that minimizes their respective routes. Let's suppose that both are members of a "Task Trading Community" whose members have agreed to use their personal agents to negotiate with other members about trading tasks – in this case delivering packages – during chance encounters.

We can formulate this scenario as a variation of the Postman domain [16] as follows:

> "Agents have to deliver sets of packages to destinations, which are arranged on a graph $G = G(V,E)$. The set V of vertices represents all possible destinations while the set E of edges represents routes along which agents can travel. Agents can exchange packages at no cost during encounters at any vertex.
>
> *Task Set*: The set of all destinations in the graph, namely V. If destination x is in an agent's task set, it means that he has at least one package to deliver to x.
>
> *Cost Function:* The cost of a subset of destination X $\subseteq$ V, i.e., c(X), is the length of the minimal path that starts at the current vertex and visits all members of X."

We use a negotiation protocol called *Product Maximizing Mechanism* (PMM) that was developed by Nash [13]. The PMM protocol is a three-step protocol. In the first step both agents disclose their task sets; in the second step each agent proposes a deal (division of tasks) that is *pareto-optimal*[1]; in the third step agents select the deal that offers both of them the most benefits. The deal that is selected and agreed upon is the deal that has the highest utility (cost savings) when the utility to agent 1 and agent 2 are multiplied.

[1] Pareto-optimal: No agent could derive more from a different agreement, without some other agent deriving less from that alternate agreement.

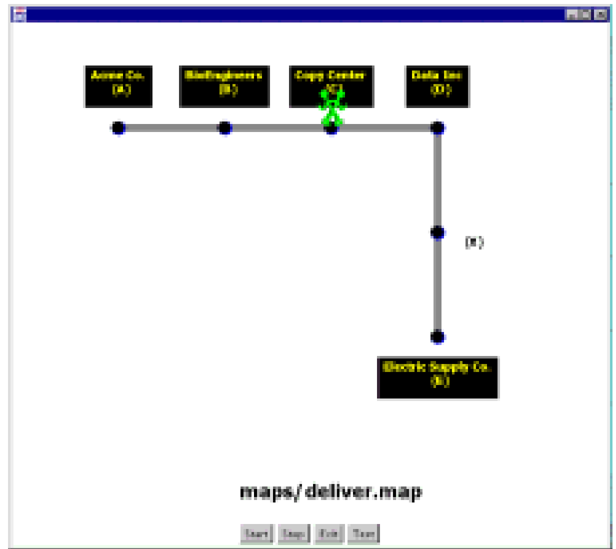

Figure 1. Graph representation of task trading domain

Let's assume two agents are negotiating at point C of the graph shown in Figure 1. Agent 1 must deliver packages to points A and D. Agent 2 must deliver a package to point A. Using the PMM system agent 1 would propose a deal that agent 1 would deliver agent 1's package to point D and agent 2 would deliver agent 1's and agent 2's packages to point A. Agent 2 would propose the deal that agent 1 would deliver agent 1's and agent 2's packages to point A and agent 2 would deliver agent 1's package to point D. The whole negotiation process looks like follows:

Initial task sets:

- Agent 1: {A, D}, cost without swapping = 4
- Agent 2: {A}, cost without swapping = 2

Proposed deals:

- Deal 1: [{D}, {A}]
- Deal 2: [{A}, {D}]

Utilities under Deal 1:

- Cost of agent 1's tasks under deal 1 = 1
- Cost of agent 2's tasks under deal 1 = 2
- Utility of deal 1 to agent 1 = 3
- Utility of deal 1 to agent 2 = 0
- Product of utilities of deal 1 = 0

Utilities under Deal 2:

- Cost of agent 1's tasks under deal 2 = 2
- Cost of agent 2's tasks under deal 2 = 1
- Utility of deal 2 to agent 1 = 2
- Utility of deal 2 to agent 2 = 1

- Product of expected utilities of deal 1 = 2

Winning deal:

- Deal 2: [{A}, {D}]

As the *Product Maximizing Mechanism* says the winning deal is the deal with the highest product of the two expected utility, deal 2 will be the winning deal. As the result, delivery driver 1 with agent 1 would deliver two packages to destination A, while driver 2 would deliver one package to destination D. An arrangement like this could save both of them a lot of time, while the end result remains the same.

Our package delivery example is an illustration of a *Task Oriented Domain* [16]: tasks can be carried out without concern about interference from other agents and each agent can accomplish his tasks without the help of other agents. On the other hand, it is possible that agents can reach agreements where they redistribute some tasks, to everyone's benefit. Negotiation is aimed at discovering mutually beneficial task distribution. The key issue here is the notion of *task*, an indivisible job that needs to be carried out.

The tasks that are negotiated can take a wide variety of forms, both physical and virtual. A trade could be "if you take my books back to the library, I'll pick up your copies from the printers" or "I'll handle all outgoing communications through my currently unused T-1 line if you process the data received." Although it is beyond the range of the initial work in this field, trades such as "I'll handle all visual data analysis if you pick up my dry cleaning" could also be useful.

In our example, we obviously made a number of simplifying assumptions that might not hold for other real-life scenarios: tasks are independent and can be carried out without interference from other agents; agents can accomplish their tasks without the help of other agents; all tasks are interchangeable and easily tradable; the cost of carrying out a task can easily be determined by both agents. This last assumption can be a problem, as determining the cost of a deal requires to compute the shortest path between a set of destinations – a process which can be computationally very expensive.

3.2 Using Deception During Negotiation

In a wearable community, the agents involved in task sharing and trading are designed and implemented by independent individuals. Maximizing personal goals is assumed to be an agent's sole motivation. It is thus possible and likely that these agents can and will use strategies in which they will lie, cheat or steal to maximize their 'owners' benefits. For example, an agent could use deception by misrepresenting its tasks or not executing tasks it agreed upon during a negotiation.

Let's consider another example in which an agent will be less than truthful about what tasks it must perform. As this is hidden information and not verifiable by outside sources this is one easy way of cheating. Let's assume that agent 1 creates a *Phantom Task* claiming tasks of {A, D, E} as opposed to the true task set of {A, D}. We then have the following situation:

Initial task sets:

- Agent 1: {A, D, E}, cost = 7
- Agent 2: {A}, cost = 2

Proposed deals:

- Deal 1: [{D, E}, {A}]
- Deal 2: [{}, {A, D, E}]

Utilities under Deal 1:

- Cost of agent 1's tasks under deal 1 = 3
- Cost of agent 2's tasks under deal 1 = 2
- Utility of deal 1 to agent 1 = 4
- Utility of deal 1 to agent 2 = 0
- Product of utilities of deal 1 = 0
- Sum of utilities of deal 1 = 4

Utilities under Deal 2:

- Cost of agent 1's tasks under deal 2 = 0
- Cost of agent 2's tasks under deal 2 = 7
- Utility of deal 2 to agent 1 = 2
- Utility of deal 2 to agent 2 = 0
- Product of utilities of deal 2 = 0
- Sum of utilities of deal 2 = 2

Winning deal:

- Deal 1: [{D, E}, {A}].

In this case the products of the expected utility will be equal. The first tie-breaking system used by PMM is the sum of the expected utility to all parties in the negotiations. This will mean deal 1 is the winning deal, with the higher sum of utility. Since E is a phantom task (information known by agent 1, but not by agent 2) the task set that will actually be performed is: [{D}, {A}].[2] By misrepresenting its initial task set, agent 1 has a true utility of 3 as opposed to a utility of 2 in the truthful example.

This example makes clear how agents can benefit from being untruthful. As protocol designers it is in our best interest to protect the honest agent. We expect that members of a wearable community will want to find a set

[2] Agent 1 must reject any deal that would pass a phantom task to the other agent.

of rules for the negotiation environment where the honest agent will perform as well as the dishonest one as this is the surest way to generate an atmosphere of trust among members – trust in the community is the single most important issue in building an effective wearable community. Without good reason to trust other members and their agents it is doubtful that people would want to join a community and cooperate with members they might not even know personally.

Rosenschein and Zlotkin [16] have shown that for some simple agent domains it is possible to define negotiations protocols in a way that 'forces' members of an agent community to adopt certain strategies, simply because they become the best strategies to advance their own goals. In particular, one can define protocols in a way that virtually prohibits agents from using deception in negotiations, because the outcome of such a lie is provably suboptimal for the lying agent (because it has to pay a higher cost or has to perform more tasks than when telling the truth.)

3.3 Building Communities

We can now rephrase our initial definition of a wearable community as a group of wearable users who a) are willing to cooperate through automated negotiation of their respective agents; b) have agreed upon a negotiation protocol for their agents; and c) have defined their own private negotiation strategies for their respective agents. A wearable community is one, in which members share a negotiation protocol. A single wearable user can belong to more than one community.

The key idea of protocols is that wearable system designers agree upon the rules of interaction, in the same way that they agree on any kind of standardization. Yet within a strategy, every wearable user can chose whatever private strategy he decides to. Members of a community do not need to know each other personally in order to engage in automated negotiation. All that is required to become a member is to implement the community protocol. These protocols could be described on special community web sites where sample agent implementations could be made available to everyone.

Good negotiation protocols create more than just fair or envy-free results. Negotiation protocols should be simple and efficient to implement, stable (so that no agent has an incentive to deviate from the protocol), and symmetric (not biased against any agent).

Two wearable agents may be expected to spend initial communication determining if they are in a common community. If they are in more than one community, i.e., they share more than one negotiation protocol, then a meta process may take place to determine the protocol that will be used. Given that different protocols support different properties, this meta-negotiation process can be key.

3.4 Additional Usage Scenarios

Automated negotiation between wearable agents does not necessarily involve trading of tasks. Negotiation is the process of several agents searching for an agreement. Agreements can be about price, meeting place, joint actions, or joint objectives.

For example, automated negotiation could take place during a swap meet, a flea market-like event where people come together in order to buy and sell rare and unusual items. One of the difficulties of swap meets is to find the person who sells the item that one is interested in. A wearable agent could identify possible trades and bid in automated auctions in competition with other agents (similar to web-based agents for online auctions [25]).

4 A Wearable Agent System

In the previous sections we discussed our idea of wearable agents and wearable communities. We now describe the design and implementation of an actual system that we use in our lab to evaluate our ideas.

As base wearable hardware platform we alternatively use a commercial Via I wearable computer with handheld monitor or a self-built wearable computer with a head-mounted display. Wireless connectivity is achieved through a Metricom wireless modem or a Lucent WaveLAN adapter. The hardware is described in more detail in [1,2,9,10].

In order to function as a wearable agent, a device must implement the following functions:

Device Discovery: A wearable agent must be able to detect the presence of other nearby agents.

Communication: A wearable agent must be able to establish a communication link with nearby devices.

Negotiation: A wearable agent must be able to perform automated negotiations.

We implemented these functions in our wearable agent system through a hierarchy of communication protocols. The architecture of the wearable system with its major software and hardware components is shown in Figure 2.

4.1 Device Discovery

We have equipped our wearable computers with near-field radio transmitters that have a maximum range of about 6-10 feet. These transmitters are connected to the parallel port and allow low bandwidth communication (14.4 baud) between devices. A beacon process that runs on each computer repeatedly sends out a unique bacon signal that can be received by near-by device. By

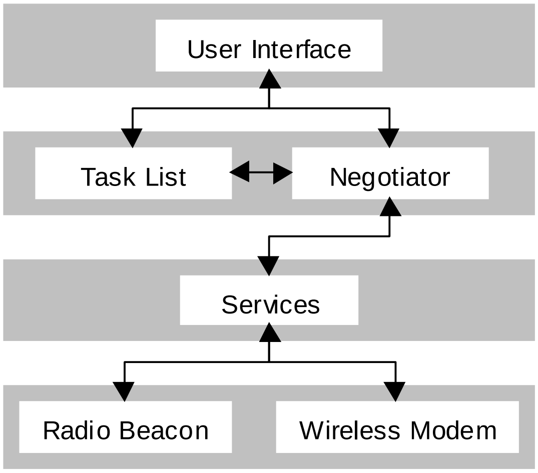

Figure 2. Wearable agent system architecture

listening for incoming signals, a device can determine which other devices are in its immediate vicinity.

We constructed the radio transmitters from radio packet controllers from Radiomatrix [20]. Figure 3 shows an early prototype without casing. The final design is much smaller, fitting comfortably in a pager case including 9V battery.

4.2 Communication

The main function of the service layer shown in Figure 1 is to establish a communication link between devices, and to give them mutual access to each other's functionality. Our solution for connecting two wearable computers devices is based on Jini, Sun's network plug-and-play architecture [19]. Jini provides simple mechanisms to plug devices together to form an impromptu community – a community put together without any planning, installation, or human intervention.

Each wearable computer implements one Jini service, which provides access to the negotiator component. Using standard Jini procedures, each device publishes its capabilities (its services) by registering them with a local *lookup service*. A lookup service is Jini's version of a service trader object. The service advertisement contains a *service handle* and an *offer descriptor*.

A device (the client) that wants to gain access to services of co-located device (the servers) has to follow a two-step procedure. The first step consists of locating the lookup services of all co-located devices. In Jini, this is accomplished by broadcasting a multicast request throughout a network. In turn, lookup services of all devices, which receive such a request answer back to let the client know that they are able and willing to provide information about services that have registered with them. The client then queries each device lookup service by supplying a *service template*. In response, it receives a collection of matching *service proxies*.[3] These service proxies, which are moved dynamically across the network, are then used to call the remote services.

Since users and their devices constantly move around in space, devices do not form long-lasting stable configurations. Thus, having discovered a service and having gained access to it through its service proxy is not a guarantee that a service is *usable*. In between the time a service was discovered and the attempt by a client to access it, the distance between both devices might be large enough so that they are out of range. If this is the case and the service cannot be reached, the service proxy is simply discarded.

The role of client and server as described above is not fixed. Each wearable device is at the same time client and server. Upon request, each device makes its service available to other devices, and each device requests access to other device's remote services.

4.3 Agent Negotiation

The wearable system has been designed for the task-trading scenario as described above. It thus contains a task list and a negotiator component. Negotiation takes place whenever two or more agents discover each other presence. Agents keep separate lists of tradable and untradable tasks. An example of a tradable task in the implemented domain is to return a book to the library. An untradable task would be to return home after a user has finished all tasks.

At the start of the negotiation, each agent dynamically determines the path to accomplish its tasks. This is currently done using a brute force iterative deepening algorithm. When agents negotiate they use their respective strategy in order to determine how they should negotiate. The expected utility of each task-swap is determined by recalculating the costs to all parties involved in the negotiations. When an agreement has been reached between agents, they ask their respective

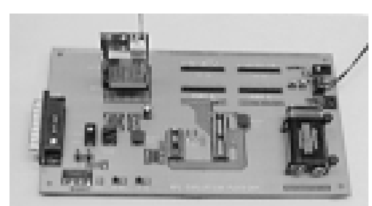

Figure 3. Radio beacon for device discovery

[3] The client filters out responses from devices that are not in the immediate vicinity using the information from the radio transmitter. This step would not be necessary if we used a Body-LAN or short-range wireless network.

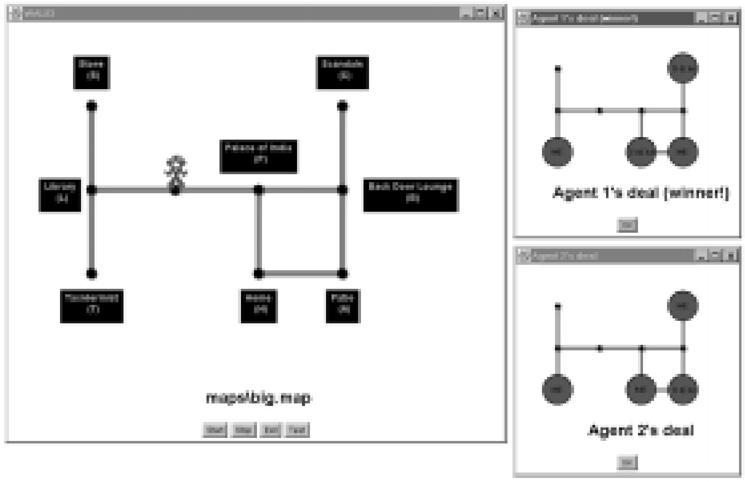

Figure 4. WALID simulator

users to approve the deal by popping up a dialog box. If both agents receive their users' ok, agents swap tasks and replace their old task-list with the new negotiated one.

5 WALID: A Simulator for Modeling Large Scale Wearable Communities

For building useful wearable communities we need to find negotiation protocols that a) are suitable for real life scenarios and b) have beneficial properties like efficiency, simplicity, stability, symmetry and fairness.

We can develop such protocols in at least two ways: On the one hand, we can use an analytic approach as Rosenschein and Zlotkin have done in [16], that is by formalizing agent domains and proving theorems about their properties. On the other hand, we could use an experimental approach. Both approaches have their own problems:

The analytic approach has the problem that it requires formal models that might not be able to capture the kind of interactions we need for real-life scenarios. On the other hand, we realized fairly quickly that it is next to impossible to design negotiation protocols from real-world experiences with wearable agents alone. First of all, since we are still in the midst of our development, there exist at best a few prototypical devices that can act as wearable agents. Second, even with a large number of functioning wearable agents, it will be difficult to get the feedback necessary to evaluate negotiation protocols and strategies.

We thus decided to build a simulator for wearable communities, called WALID. WALID is an interactive system for studying the behavior of a large number of cooperative wearable agents in a wearable community. The current version of WALID is limited, but powerful enough to simulate the task-trading scenario; other domains will be added in the future.

With WALID we can easily vary the number of agents in a wearable community, the negotiation protocol, and the private negotiation strategy of each agent. Similarly, we can define initial task lists and initial location for each agent. A simulation consists of agents moving around in space according to their tasks, and engaging in negotiation when two or more of them meet. As a result of the simulation we obtain a record of performed negotiations and the values achieved by each individual agent in these negotiation.

Agents in the WALID world operate in a customizable map space. Maps are graphs with edges and vertices and define possible movements of agents. WALID restricts the edges follow the cardinal directions (North, South, East and West.) All edges in WALID maps are of uniform length 1. The length requirement does not restrict what graphs are possible, as dummy vertices may be used to create edges of any dimension.

Figure 4 shows two agents during an encounter in WALID. Both agents have a set of destination goals that define the locations they have to visit. The large window (left) shows the current map they are operating under with their respective location. The two smaller windows (upper & lower right) show the deals proposed by each agent in the negotiation. The upper right window in Figure 4 shows the new set of tasks after the negotiation has concluded.

WALID maps and task-lists are created using XML as file format. This allows for easy creation and human readability, and leaves open the potential for expansion. The WALID simulator is written in Java for cross platform use and ease of modification. The source code is available at: http://www.cs.uoregon.edu/research/wearables/walid/source/.

As the agents in this task are wearable users we have also implemented a *Cost of Negotiation* (CoN). CoN is a constant value subtracted from the utility of a new task set in step 2 of PMM. This value relates to the human cost of changing the wearable users task set. In our package deliver example, tasks relate to traveling to physical locations. By adding a CoN value we prevent the wearable user from pointlessly trading tasks and spinning in place.

There are several areas for future work on WALID. We want to implement and test other protocols (for example Sum Maximizer), and compare them to PMM, as PMM assures an efficient solution but not a globally optimal one. The WALID simulation values must be compared to the values determined in real world wearable testing. Finally we hope to extend the simulator to other domains, both task-oriented and non-task-oriented.

6 Related Work

The idea of using wearable computers for cooperation and collaboration has been around at least as long as wearable computers have become feasible. In the recent past, systems have been developed both for co-located and remote collaboration. Among the first systems were several designed at Carnegie-Mellon University [8,12,17]. Similar systems have been built and evaluated at the University of Washington [3,4] and by our own research group [1,2,10]. While [3] is one of the few systems designed for co-located collaboration, it is still a sophisticated, yet traditional videoconference system intended for visual and audio communication.

Quite different from these applications is the GroupWear system designed at MIT [5]. GroupWear is a computationally augmented nametag capable of providing information about relationships between two people in a face-to-face conversation. GroupWear uses infrared transmitters to exchange personal profile information during face-to-face conversations. In contrast to our work, however, the interaction between nametags is rigid and does not involve negotiation.

Proem [9], a system developed at our lab, is the precursor to the work described in this paper. Proem is based on the notion of *personal profiles*, which are exchanged whenever two or more wearable users encounter each other. It is very similar to the system we described here, in that it supports cooperation during face-to-face encounters. However, it lacks the concept of negotiation, and instead only supports the simple exchange of personal information. It is then up to the individual user to decide how to make use of this information.

Our research on communities is related to the notion of *community computing*, a concept that was coined by Toru Ishida [6,7] at Kyoto University. Community computing deals with the creation, maintenance, and evolution of social interaction in communities. Community computing is intended to support more diverse and amorphous groups of people than traditional groupware, and supports the process of organizing people who are willing to share some mutual understanding and experiences. In other words, community computing focuses on the group formation stage. In a similar approach, Hattori et al [26] describe a multiagent system for supporting network communities.

Many of the theoretical foundations of game theory and cooperation have been formulated by John Von Neumann and Oskar Morgenstern [15], and John F. Nash in the late 40th [13,14]. For most of our treatment of agent cooperation, we follow Jeffrey S. Rosenschein and Giliad Zlotkin [16]. Their approach is somewhat unique in that they apply analytical techniques and modeling methods from the world of game theory and decision analysis to the dynamic organization of autonomous intelligent agents. Their rigorous analysis of agent systems provides a formal basis for research in agent interaction. However, they are solely concerned with the interaction between machines without interference of human intervention. The domains they analyze are necessarily simple compared to interactions that we would like to create in a wearable agent community.

7 Conclusion and Outlook

The notion of Cyborgs, communities of Cyborgs, and Collective Intelligence has been around at least as long as wearable computers have become feasible. Yet, today's wearable users largely live a lonely and disconnected life. Other than email, interaction with fellow Cyborgs is limited to the old fashioned way – through unmediated and un-augmented face-to-face conversation. We are working on the creation of communities of wearable users where opportunistic personal agents embedded in wearable computers cooperate on behalf of and in the interest of their 'owners'. We are particularly interested in how this technology can be used to enable goal-directed cooperation during encounters of people with selfish and conflicting goals, such that cooperation leads to mutually beneficial results.

In the preceding sections, we have done three things: first, we put forward a theoretical framework for building communities of cooperating wearable agents. Second, we described the initial implementation of a wearable agent device; finally we described a simulator that we use to develop and analyze negotiation protocols and strategies.

There are many areas for future work. The first step towards building cooperating wearable communities is the creation of the actual hardware and software systems that are able to establish communication with other wearable device in an ad-hoc manner. Future development in wireless networking (short-range wireless networks and body-LANs) will certainly contribute to this.

Much more unclear is how to set up a network of rules that guides the interaction of these agents. We believe we made a first step towards this goal by defining properties and characteristics of agents and their interaction. This includes notions like goal-directed and opportunistic behavior and the distinction between negotiation protocols and strategies (which we borrowed from [16]).

The development and evaluation of such protocols is made difficult by the highly distributed nature of a wearable community and the fact that agent negotiation becomes part of the social environment of its owner. We thus believe that simulating large-scale wearable communities on the computer is the only feasible way to derive agent protocols for wearable communities.

Finally, it is important to note that with the work described herein we do not propose replacing human interaction with agent interaction. The task of personal agents in a wearable community is to identify opportunities for possible cooperation and to propose them to their users. Whether or not two individuals ultimately want to cooperate depends on a careful consideration of potential benefits and risks based on their own personal judgment.

We are now pursuing continued research in the directions opened up in this paper, including the development of new negotiation protocols and the improvement of the WALID simulator.

References

[1] Bauer, Martin; Heiber, Timo; Kortuem, Gerd; Segall, Zary. *A Collaborative Wearable System with Remote Sensing*. Proceedings Second International Symposium on Wearable Computers, Oct 19-20, 1998, Pittsburgh, PA

[2] Bauer, Martin; Kortuem, Gerd; Segall, Zary. *"Where Are You Pointing At?" A Study of Remote Collaboration in a Wearable Videoconference System.* Proceedings Third International Symposium on Wearable Computers (poster), Oct 18-19, 1999, San Francisco, CA.

[3] Billinghurst, M.; Weghorst, S., and Furness, T. *A. Wearable Computers for Three-Dimensional CSCW*. First International Symposium on Wearable Computing; 1997; Boston, MA.

[4] Billinghurst, M; Bowskill, M.; Jessop, M., and Morphett, J. *A Wearable Spatial Conferencing Space*. Proceedings of the Second International Symposium on Wearable Computers; 1998 Oct; Pittsburgh, PA.

[5] Borovoy, R., Martin, F., Vemuri, S., Resnick, M., Silverman, B. and Hancock, C. (1998b) Meme Tags and Community Mirrors: Moving from Conferences to Collaboration. In *Proceedings of CSCW '98*, ACM Press.

[6] Ishida, Toru, "Towards Communityware," *International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-97)*, Invited Talk, pp. 7-21, 1997.

[7] Ishida, Toru Ed., *Community Computing: Collaboration over Global Information Networks*, John Wiley and Sons, 1998.

[8] Kraut, Robert E.; Miller, Mark D., and Siegel, Jane: *Collaboration in Performance of Physical Tasks: Effects on Outcomes and Communication*. Computer Supported Cooperative Work '96 Cambridge, MA, USA: ACM; 1996.

[9] Kortuem, Gerd; Segall; Zary; Cowan Thompson, Thaddeus G. *Close Encounters: Supporting Mobile Cooperation Through Interchange of User Profiles* . 1st International Conference on Handheld and Ubiquitous Computing (HUC), September 1999, Karlsruhe, Germany.

[10] Kortuem, Gerd; Bauer, Martin; Heiber, Timo; Segall, Zary. *Netman: The Design of a Collaborative Wearable Computer System*. ACM/Baltzer Journal on Mobile Networks and Applications (MONET), 1999.

[11] Rhodes, B. *The Wearable Remembrance Agent: A System for Augmented Memory*. Proceedings of the First International Symposium on Wearable Computers (ISWC'97), Cambridge, MA, October 1997.

[12] Siegel, J., Kraut, R.E.., John, B.E., Carley, K.M. *An Empirical Study of Collaborative Wearable Systems*, CHI Companion 1995, Denver, Colorado, 312-313.

[13] Nash, John F.. The Bargaining Problem. *Econometricia*, 28:155-162, 1950.

[14] Nash, John F. Two-person cooperative games. Ec*onometricia*, 21:128-140, 1953.

[15] Von Neumann, John; Morgenstern, Oskar. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, 1947.

[16] Rosenschein, Jeffrey S.; Zlotkin, Giliad. *Rules of Encounter, Designing Conventions for Automated Negotiation among Computers.* MIT Press, Cambridge Massachusetts, 1994

[17] Siewiorek, D.; Smailagic, A.; Bass, L. ; Siegel, J.; Martin, R. and Bennington, B. *Adtranz: A Mobile Computing System for Maintenance and Collaboration.* Proc. of the 2nd International Symposium on Wearable Computers; 1998 Oct; Pittsburgh, PA

[18] Zimmerman, T. G. *Personal Area Networks: Near-field Intrabody Communication*, IBM Systems Journal, Vol. 35, No. 3&4, 1996

[19] Sun Microsystems. http://www.javasoft.com/products/jini

[20] Radiomatrix Inc. http://www.radiomatrix.co.uk

[21] Huhn , Michael N.; Mohame, Abdulla; *Benevolent Agents.* IEEE Internet Computing, March-April 1999, p.96-98.

[22] *Bluetooth* Consortium web site, http://www.bluetooth.org.

[24] Schneider, Jay; Suruda, Jim; Fickas, Steve. *Modeling Agent Negotiation in a Task Oriented Domain*. Proceedings Third International Symposium on Wearable Computers (poster), Oct 18-19, 1999, San Francisco, CA.

[25] P. Maes, R. Guttman and A. Moukas. *Agents that Buy and Sell: Transforming Commerce as we Know It. Communications of the ACM*, March 1999 Issue.

[26] Hattori, Fumio; Ohguro, Takeshi; Yokoo, Makoto; Matsubara, Shigeo; Yoshida, Sen. *Socialware: Multiagent Systems for Supporting Network Communities. Communications of the ACM*, March 1999 Issue.