CoolAgent: Intelligent Digital Assistants for Mobile Professionals – Phase 1 Retrospective

Martin Griss, Reed Letsinger, Dick Cowan,
Michael VanHilst, Robert Kessler[1]
Software Technology Laboratory
HP Laboratories Palo Alto
HPL-2002-55 (R.1)
July 9th , 2002*

personal assistants, agents, JADE, CoolAgents, mobile professional

This report summarizes the key goals, vision, and accomplishments of research in HP Laboratories on Assistant and Agent technology for me-centric computing. The primary focus of our work is in the architecture and implementation of intelligent assistants for mobile professionals and me-centric, peer-to-peer computing. We are investigating the possibilities and consequences of agent-mediated interaction between users, appliances, services and information. During 2001, we completed a major release of our JADE and Bluestone based meeting arranger multi-agent prototype, which we call CoolAgent Phase 1. The report summarizes the accomplishments and learnings of our completed CoolAgent partnership release and experiments and demo in the mobile professional, distributed meeting arranger domain.

* Internal Accession Date Only
Approved for External Publication
[1] University of Utah, Salt Lake City, UT, 84112

CoolAgent: Intelligent Digital Assistants for Mobile Professionals - Phase 1 Retrospective

Martin Griss, Reed Letsinger, Dick Cowan, Michael VanHilst, Robert Kessler*
Agents for Mobility Department (and *University of Utah)
Software Technology Laboratory
Hewlett-Packard Labs
1501 Page Mill Road
Palo Alto, CA 94304 USA
June 2002

ABSTRACT

This report summarizes the key goals, vision, and accomplishments of research in HP Laboratories on Assistant and Agent technology for me-centric computing. The primary focus of our work is in the architecture and implementation of intelligent assistants for mobile professionals and me-centric, peer-to-peer computing. We are investigating the possibilities and consequences of agent-mediated interaction between users, appliances, services and information. During 2001, we completed a major release of our JADE and Bluestone based meeting arranger multi-agent prototype, which we call CoolAgent Phase 1. The report summarizes the accomplishments and learnings of our completed CoolAgent partnership release and experiments and demo in the mobile professional, distributed meeting arranger domain.

Keywords: Personal Assistants, agents, JADE, CoolAgent, mobile professional.

1 Introduction

The primary purpose of this report is to describe the design and development methods used to construct a medium-scale multi-agent prototype, and to capture some of the lessons learned from this experience that may help future developments of multi-agent systems. Our long-term objective is to understand how to effectively construct intelligent assistants for mobile professionals. In this initial testbed and demonstration, we focused on the specific task of providing support for distributed meeting arrangement. This is primarily a scenario for envisioning features and issues, and a driver for prototyping and evaluation of multi-agent technology and testbeds.

A mobile professional must play a variety of changing and intertwined work-related and personal roles, pursue multiple ongoing tasks, and adapt to a changing environment of appliances and services. People in many different situations face the need to juggle roles, adapt to new contexts, and filter information: mobile sales organizations, support personal, working parents, convention floor attendees, shoppers in a mall, and visitors to theme parks, to name a few. We imagine the user employing an intelligent, customizable, proactive Personal Assistant to improve productivity and reduce friction, simplify the use of appliances and services, help in changing appliances and roles, and aid the user in making the most effective and convenient use of these dynamic environments. The Personal Assistant (PA) can help the user in several ways: the user can delegate tasks to the PA; the PA can monitor progress of the user's tasks and detect opportunities or threats to those tasks; the PA can filter in-flowing information and can route notifications to appropriate devices.

Over the first half of 2001 we collaborated with several partners from different organizations within HP to design and develop a sample workplace productivity solution and agent testbed. In this testbed, we deployed many agent-based intelligent personal, team and service assistants that together help the mobile professional arrange and manage distributed meetings. Some of our current and planned efforts were inspired by Patty Maes' visions of agents that reduce work and information overload [Maes, 1994]. The testbed used the JADE agent toolkit [Bellifemine et al., 1999] and the HP Bluestone application server [Bluestone 2001], as well as a variety of pre-existing calendar and meeting services.

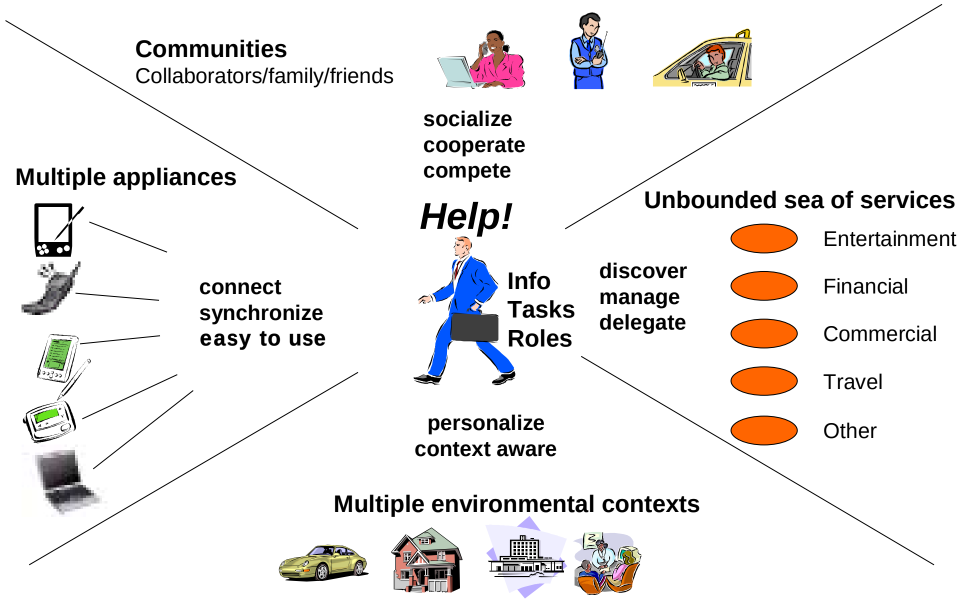
Earlier visions, scenarios and prototypes involving Zeus agents [Nwana et al., 1999], e-commerce and CoolTown [Kindberg et al., 2000] significantly influenced our thinking, and motivated our choice of domain and technology. These earlier scenarios and prototypes are described in our papers on e-commerce simulations, e-commerce and workflow, shopping mall assistants and intelligent agents [Griss, 2000; Chen et al., 2000; Griss and Letsinger, 2001; Fonseca et al., 2001; Fonseca et al., 2001a], similar in spirit to the agent-based shopping agents described by Patty Maes and colleagues [Chavez and Maes, 1996;. Guttman et al., 1998; Maes et al., 1999].

2 Vision

2.1 A complex world of people, appliances, and services

In the world that we imagine, a mobile user interacts with his environment through one or more web-enabled appliances. Some of these appliances are mobile (such as a PDA, badge, cell phone, or pager), and others are affixed to a place or a thing (a PC, devices in a CoolTown instrumented meeting room or mall, or Web-Kiosks and public information portals). At any time, a user might have a primary appliance that he carries with him, and additionally have temporary access to a dynamically changing set of other mobile or fixed appliances. (Figure 1)

Figure 1: Modern Life Is Connected & Complex



The user wants to interact with other users, and a vast array of dynamically changing information and computational services. For example, in the distributed meeting arranger and manager scenario we are using, a user may need to interact with other users, their calendars, their preferences, room bookers, teleconferencing services, chat rooms, NetMeeting, and presentation services. This dynamically changing, highly interconnected world is complex and requires the user to deal with many details and hassles.

2.2 Intelligent assistants

To reduce complexity and to mediate between multiple users, appliances, and services, we imagine that the mobile professional has a Personal Assistant (PA) that provides intelligent assistance, wherever he goes, whatever he does (Figure 2).

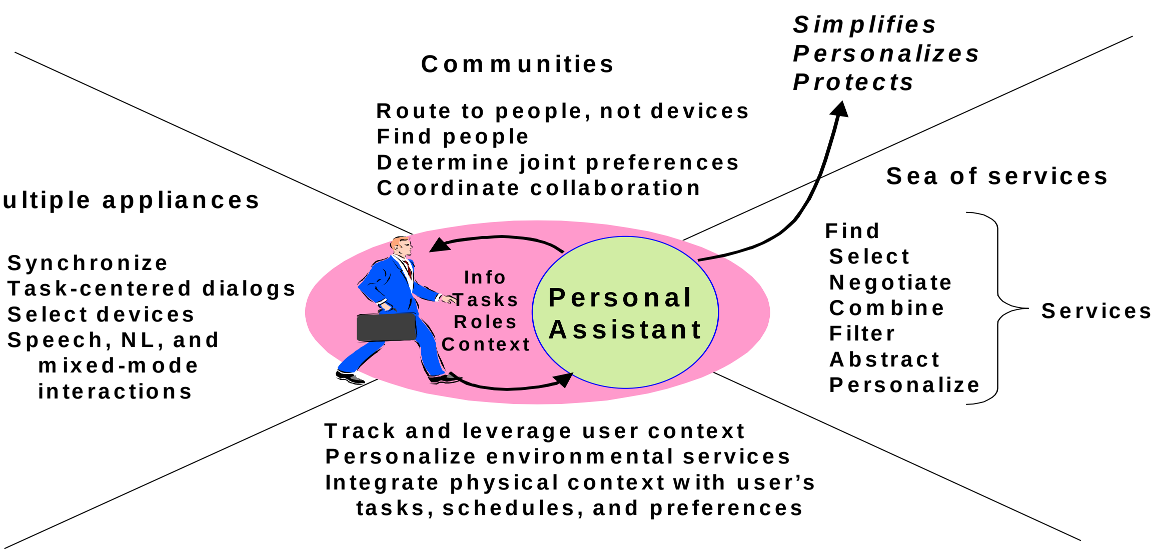
Figure 2: Vision: “Intelligent assistance,anywhere, anytime, for any task”

Imagine: ***intelligent software assistants that work proactively for me to***

- pursue tasks that I have delegated
- vigilantly monitor changes in my situation and tasks
- notify of progress and concerns
- make recommendations
- negotiate on my behalf

A Personal Assistant is the nexus of context-based, personalized "me-centric" computing (Figure 3) [Hayes Roth 2001]. How the Personal Assistant behaves depends on the user’s preferences, policies, goals, priorities, and schedules, as well as preferences and policies of any groups the user belongs to, and information about the user’s current location and activities. Preferences and context will also determine how information is coded and filtered for different appliances, and how information is partitioned, secured and made anonymous for interaction with other users and services.

Figure 3: Personal Assistant

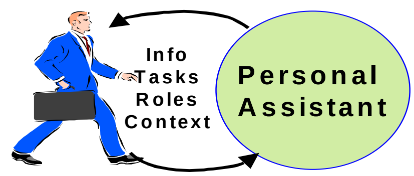


Preferences and context may dictate which appliance and appliance mode to use under various circumstances. The Personal Assistant will help in maintaining a *seamless session* as a dialog with an appliance is moved to another more appropriate appliance (e.g., voice directions on a phone switched to a map on a Kiosk). Context and preferences will determine the degree of delegation or autonomy the user chooses to give to the Personal Assistant to act on his behalf. The user will be able to specify how much autonomy the Personal Assistant should take when it negotiates and concludes transactions while shopping, or when it rearranges lower priority meetings in order to free time for a more important scheduled activity (Figure 4). Natural communication and ease of use are of paramount importance.

Figure 4: Intelligent Assistants

At work, at home, or on travel, personal assistant agents could

- schedule appointments and events
- coordinate with colleagues and friends
- arrange travel
- shop for goods and services
- manage my email and voicemail
- find information, people, locations
- adapt to my needs, learn from experience



….. and so simplify and enhance my life

The key roles of a PA are summarized in Table 1 and key requirements for a PA are summarized in Table 2.

Table 1 – Key Roles of a Personal Assistant

- Simplify the use of appliances
- Personalize services
- Reduce the user's 'interaction costs' with e-services
- Protect the user from the intrusion of unwanted information and requests
- Serve as a trusted repository of personal information
- Help coordinate complex tasks involving multiple services and people

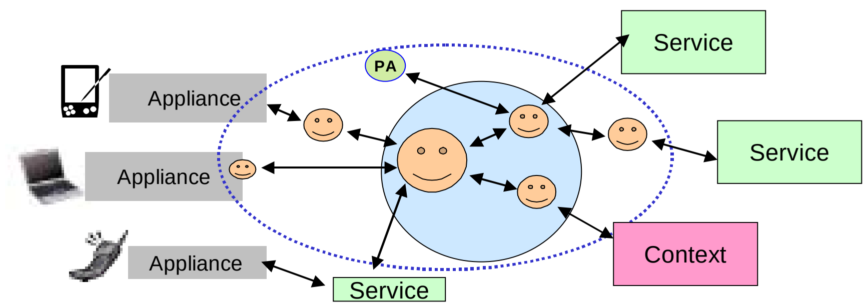
Table 2 – Key Requirements for a Personal Assistant

- **Easy to use -** easy to learn to use
    - easy to use once learned
    - easy to personalize
- **Trusted -** secure management of personal information
    - flexible privacy policies
    - easy to understand
- **Configurable -** from very rudimentary to highly sophisticated, specialized assistants for particular roles with adjustable delegation and autonomy

2.3 Agent-mediated, distributed peer-to-peer system

We implemented our Personal, Team and Service Assistants using distributed software agent technology. We created a Personal Assistant using a set (or *society*) of software agents that work together. Some work closely with or on the appliances, while others are more closely allied to personal services, such as my calendar or email (See Figure 5).

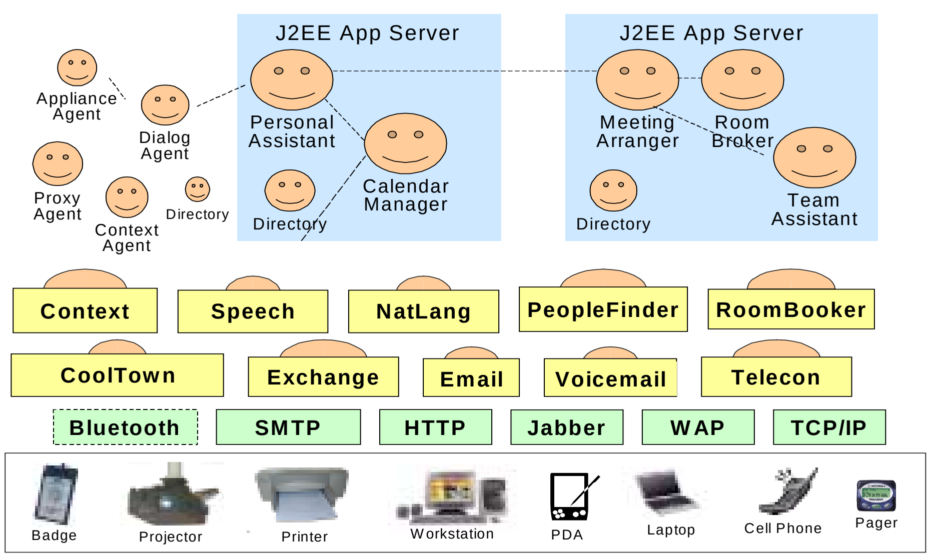
Figure 5: A Personal Assistant is a Community of Agents



- PA collects agents and services that have access to and are customized by personal information
- PA's agents may run on different devices, including mobile appliances
- One agent is the primary point of contact to the user and the outside world, and coordinates the behavior of the other agents and services

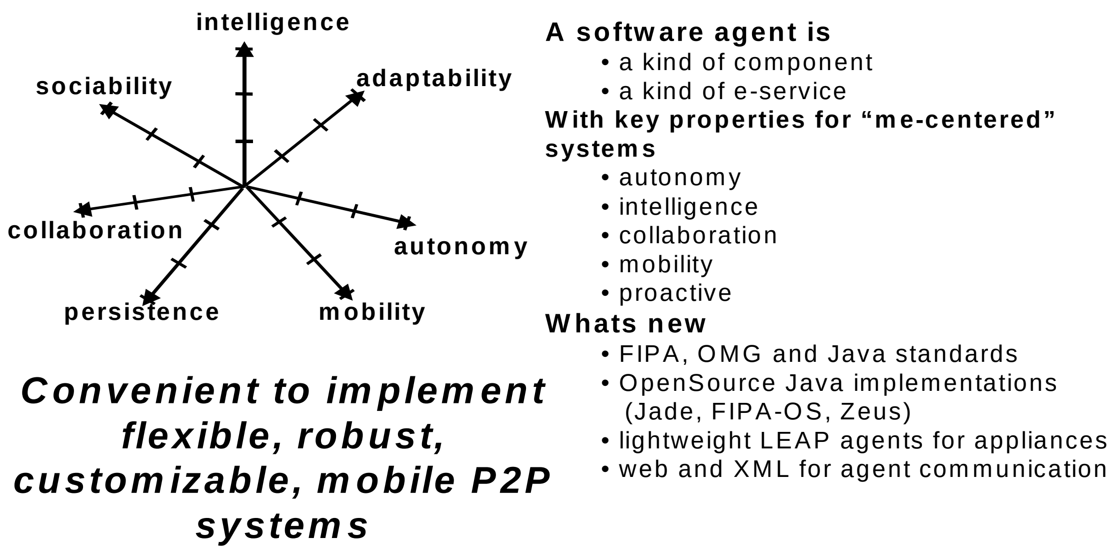
In addition, we used a variety of task-oriented or service-oriented agents to provide a uniform "dynamic service-oriented interface" to various services (Figure 6).

Figure 6: Multi-Agent Architecture



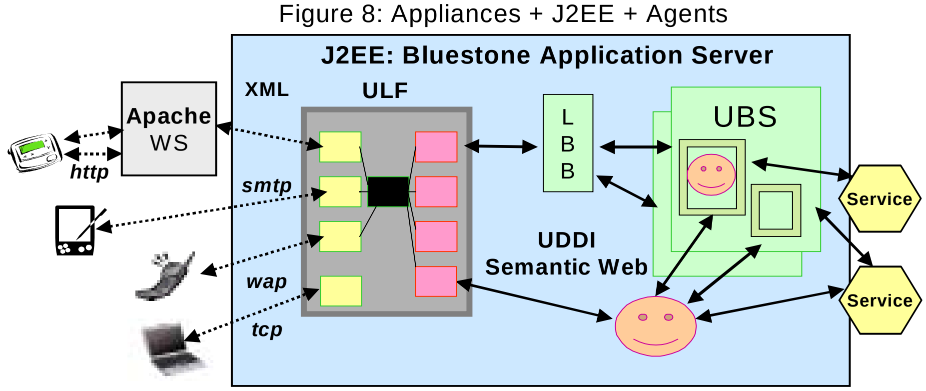
An agent is a special kind of component, ideally tuned for loosely coupled, dynamic systems. It displays some degree of autonomy, intelligence, dynamic peer-to-peer computing abilities, and other attributes summarized in Figure 7. We are using the JADE agent system, a Java open source, FIPA-compliant multi-agent system [O'Brien and Nicol, 1998; Bellifemine et al., 1999]. Agents communicate with each other using structured messages in Agent Communication Language (ACL). ACL messages are similar to KQML messages [Finin et al., 1997], and support asynchronous message exchange, controlled message flow based on protocols, and rich semantic content. The content of our ACL messages is defined by the CoolAgent Ontology [Sayers and Letsinger, 2001]. Agents can also register themselves and their capabilities with both "white page" and "yellow page" directories for lookup by name and by capability from anywhere in a multi-machine environment.

Figure 7: Software Agents



The agents and services that make up an assistant may be distributed, and some might live on appliances, while others are hosted on specially designed, secure agent servers (see Figure 5). Agents and services running on the servers are managed by special containers that have been integrated with the HP Application Server platform (Figure 8).

Figure 8: Appliances + J2EE + Agents

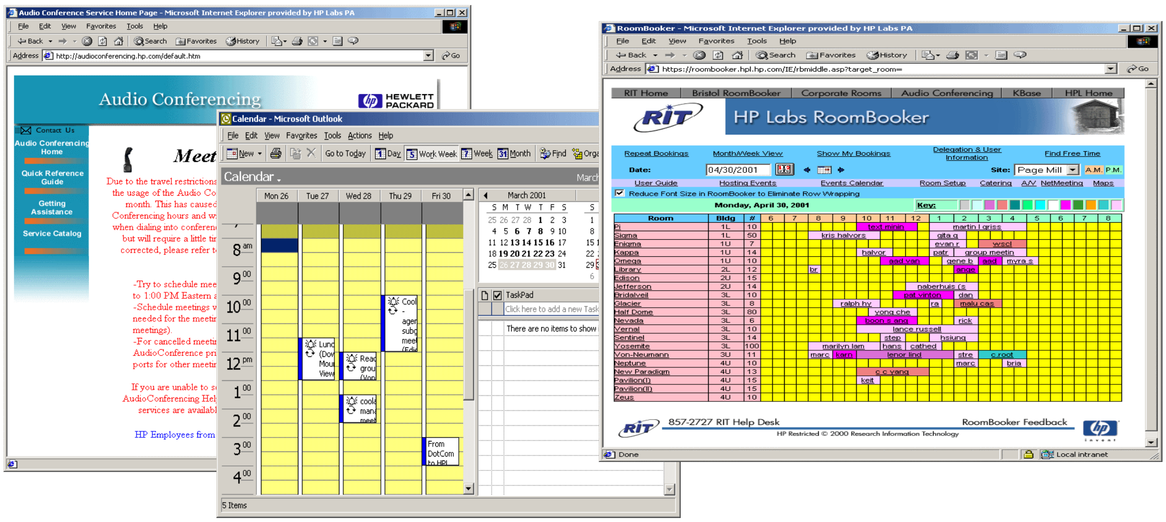


- Handles multiple protocols, ports, XML format conversions
- Provides security, configuration, management, scalability services
- Agent platform becomes Bluestone CSF container/service

3 Scenario - distributed meeting arranger

The current way of setting up meetings is to use several, independent interactions with three or more different applications, each with its own interface, and little awareness of each other. For example, as shown in Figure 9, we might use Outlook/Exchange to access the calendar, HP Labs RoomBooker to reserve a room, and MeetingPlace to reserve a teleconference line. We might also send email to others, or use the phone to see if they are available. These steps are time consuming and error prone, particularly when a meeting has to be changed or cancelled, or if an ideal date and location have to be negotiated. Sometimes we go through these steps and applications ourselves, but if we are lucky, we can delegate this task to our administrative assistant, who can handle all the details.

Figure 9: Meeting Management – the old way



They all have different interfaces
There's no "intelligence"

They don't share information
Tedious, error-prone, slow

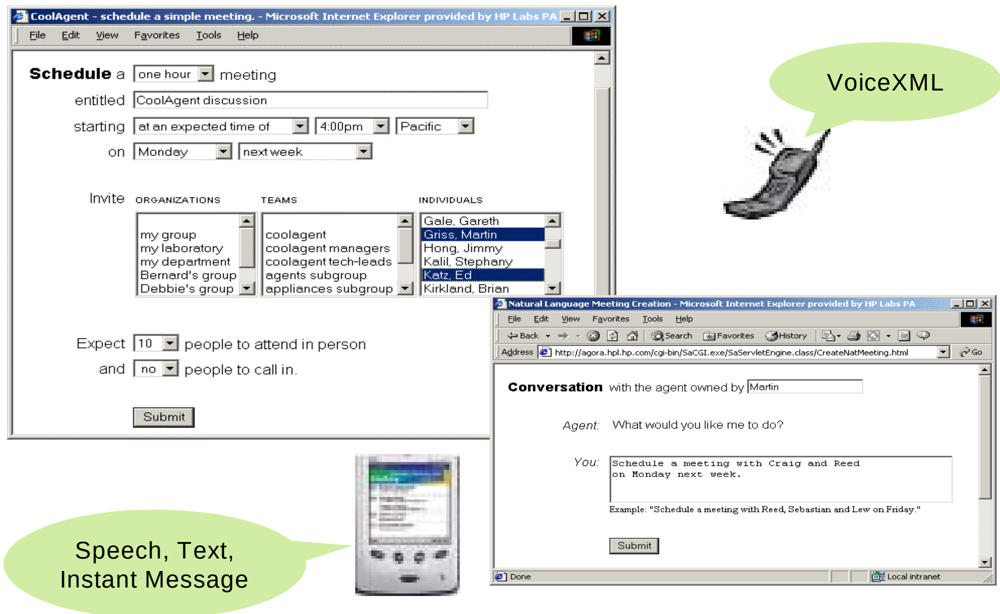
In our CoolAgent prototyping effort, we imagine (literally) saying to our Personal Assistant "*set up a meeting with Reed and Dick next week*", or using a simple "one button" meeting form, to have our PA interact on our behalf with the various applications and other meeting participants (Figure 10).

This request could be submitted from wherever I am. My PA will work on my behalf, even when I am travelling, to arrange the meeting. As shown in Figure 10, we have explored several levels and modes of interaction with the PA to set up meetings from browsers on a PC, Laptop or PDA. We have also explored the use of spoken natural language from a wireless HP Jornada, using a restricted English subset, and more recently the use of VoiceXML for a telephone interface. We are also prototyping an instant messaging interface (using Jabber).

At this point, the user can communicate with the PA using email, voice, telephone or a browser. The PA can communicate back to the user using the browser, email, voicemail or pager, as appropriate. Some of our work on context and preferences allows the PA to choose to communicate using an appropriate appliance and mode based on my location, my preferences, the urgency and other contextual items.

The meeting agent will then notify my PA, and the PA of the other participants of the agreed meeting. My PA will then choose how to notify me, using email, voicemail, or some other mode. These messages are formatted appropriately for the device to which they are sent, and some care is taken to reduce the message size while loosing as little information as possible. For example, the system will use "Monday, April 7" in a message to a desktop machine, and "4/7" in a message to a pager. Figure 11 shows some of these emailed notifications; voicemail or short messaging modes are similar.

Figure 10 - "One Button" Meetings the new Way



As summarized in Figure 12, my Personal Assistant will negotiate with the CoolAgent community assistant and the PAs for other team members. A meeting agent will be created for this meeting, so that subsequent interactions to refine or rearrange the meeting have a point of presence to interact with the meeting as a "Thing." The Meeting agent will ask my PA to check my calendar; it can consult broker agents for teleconference lines and room availability. It can take note of the team members' locations and schedules scattered across the Bay Area, Fort Collins and France, take note of preferences and priority of conflicting meetings, time zones, etc. Only my PA knows what calendar(s) I use, if any. My PA is responsible for answering questions about my availability for certain meetings, and not about my schedule in general. This allows rules, context and other information to be used to decide if I should attend the meeting, and with what priority I can commit.

Figure 11: Arranger and attendee email

…. an email or voicemail arrives in your in-box:

"Regarding 'CoolAgent Discussion'.
I have arranged that on Monday, April 7th at 4:00 PM PDT for one hour in Kappa, building 1-Upper, 1501 Page Mill Rd, Palo Alto.
I received responses for that from Ed(possibly), Reed(certain), and yourself."

…. At the same instant, the attendees also receive email or voicemail:

"You are scheduled to attend a meeting: 'CoolAgent discussion' on Monday, April 7th, at 4:00PM Pacific Daylight Time for one hour in Kappa, building 1-Upper, 1501 Page Mill Rd, Palo Alto.

Figure 12: Behind the scenes

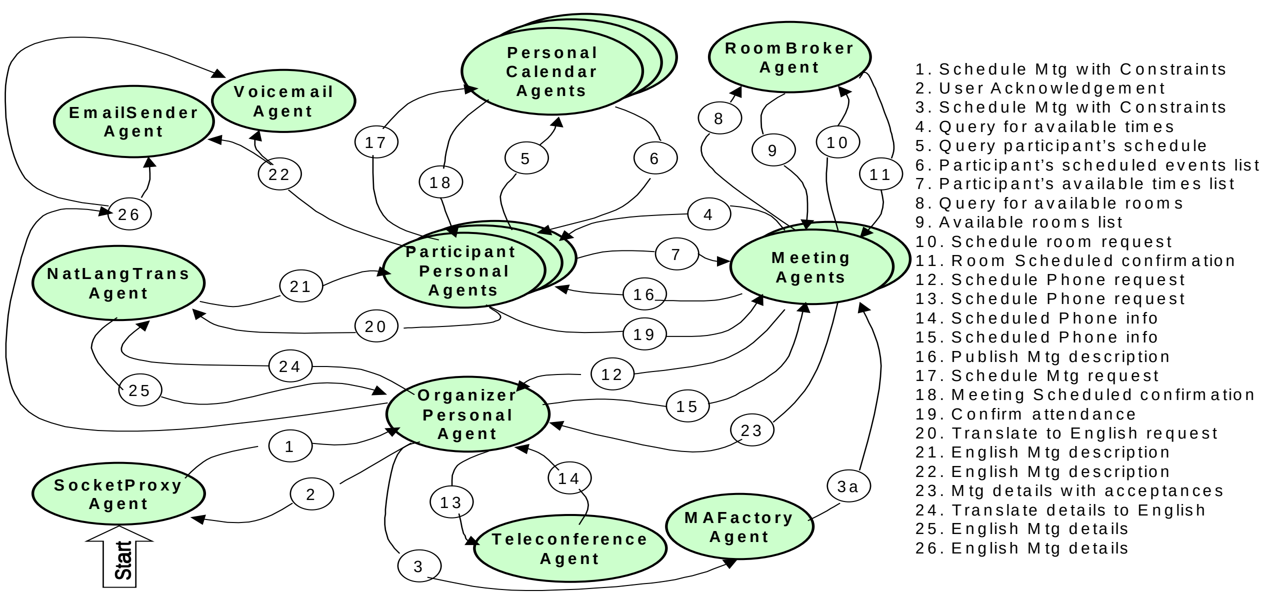
1. The form was sent to your personal assistant.
2. It asked the system to make a meeting agent.
3. That meeting agent:
   - Checked the personal assistants of all invitees to find possible times
   - Checked with Roombroker to find rooms
   - Asked the personal assistants to rate the possible rooms/times based on their owner's preferences
   - Selected a room and time.
   - Booked the room
   - Asked each personal assistant to schedule the event in their owners' calendars.
   - Notified your personal assistant, and those of all attendees, that the event had been scheduled.
4. The personal assistants then notified their owners.

My PA will set up the meeting for me, using my calendar if appropriate, and either my PA or the meeting agent will reserve rooms and teleconference lines. It is important to note that there is no central, global calendar. The agents communicate "peer-to-peer", and each user and the user's PA can use different calendaring tools and interfaces.

This flexible design makes it relatively easy for me to have a PA with quite distinct and different behavior from yours. Mine might access an Outlook calendar; yours can access your Palm desktop, while other PA's might simply refuse to have meetings with certain people in the mornings, or on Friday afternoons in general.

Figure 13 shows the complex set of peer-to-peer message exchanges between the various agents to set up a typical meeting. This does not show the many exceptions that may arise, and all the alternatives that might need to be explored.

Figure 13: Interaction to schedule a meeting



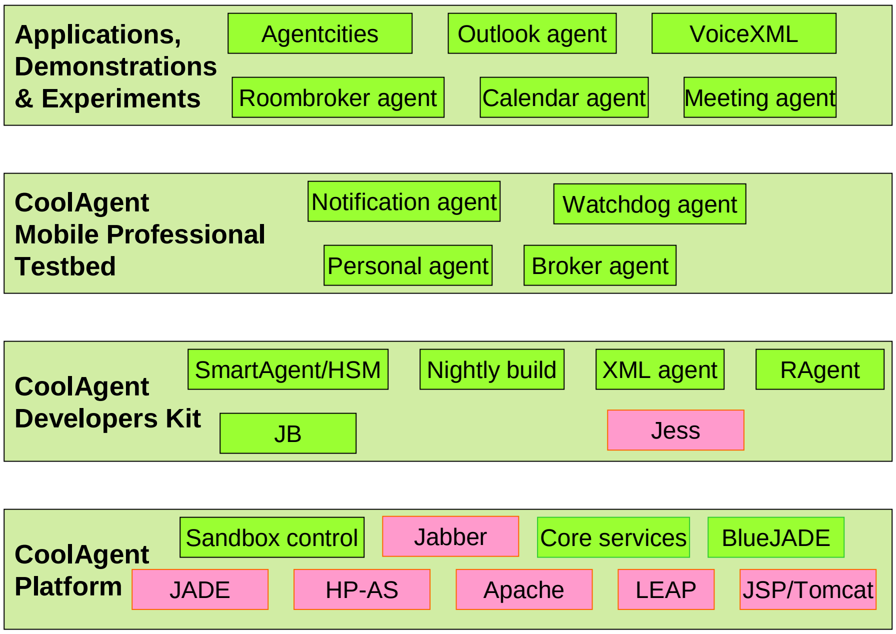
4 Technology

One of our goals was to prototype a usable testbed, and so we are using as many real-world products and existing HP services as is feasible, with agents serving as "intelligent, flexible mediators." This is part of a desire to have an "industrial strength" agent-platform.

4.1 Testbed technology

Figure 14 summaries the layers and components in our testbed.

Figure 14: Layered, Modular Technology



We are using product software such as HP's Bluestone Application Server, as well as integrating some open source software (Apache, JADE agents, XML parsers). We are extending the various open source Java-based agents, rule systems and fuzzy logic packages. We are using Apache, JSP and XML for our main Web-servers.

We are integrating a variety of HP IT applications or prototypes (Exchange, Roombooker, Reservit, VIEW, MeetingPlace, Dragon Systems and WAP). Many of these systems are web-based, but most were not designed for easy programmatic or XML access. Since they do not provide a uniform "e-services" interface, we have had to develop many idiosyncratic wrapper-agents. We have used HPL Roombooker, as well as a corporate voice-email-wireless portal (VIEW) to provide an interface to the Microsoft exchange server and to the Octel voicemail system.

We have created interfaces for several appliances, such as Jornadas, phones (WAP), and desktops. The Bluestone totally e-mobile product (ULF, LBB and UBS) provides a base for seamless, multiple appliance access, and information transcoding and compression. This work has continued to produce a much tighter integration between JADE and Bluestone, called BlueJADE [Cowan et al, 2001].

We have invented a new ontology for meeting management [Sayers and Letsinger, 2001]. In an advance over previous systems [Dawson et al., 1998], this supports meetings that are both spatially and temporally distributed. It also supports the "fuzziness" which is a natural consequence of arranging meetings among real people in the real world. For example, it allows you to express the concept "I'd prefer to meet with the whole team in Palo Alto on Friday morning, but it would be acceptable to meet with just the team leads in Mountain View on Wednesday".

"Intelligence" is provided using rule systems, fuzzy logic, planners and schedulers to create a flexible and natural system. We have just begun to exploit their capabilities in more than a rudimentary way, and will now focus on significantly increasing the decision making power.

We have prototyped the use of restricted continuous speech for a context dependent memory assistant, and have extended that to provide a speech and natural language interface to our system. We have also implemented a simple web-based profile editor.

4.2 Relationship to CoolTown

The Personal and Community Assistants and agents build on and extend the CoolTown "People, Places and Things" metaphor [Kindberg et al., 2000] in a variety of ways. For example, in our model, a meeting room "place manager" is represented by an agent, which has its own web presence. The user can interact with the meeting room agent using either a web interface or the Agent Communication Language. The meeting room agent also acts as a local "facilitator" (using the yellow and white pages) with which appliances, services and meetings can register upon discovery. Likewise, a meeting itself becomes a Thing, accessed in a variety of ways (e.g. via Web presence or agent communication language), etc.

We also share a strong belief in the role of context-aware computing for the me-centric, peer-to-peer vision. The PA can potentially use preferences, schedules, available appliances, tasks, goals and locations as elements of context to shape the interaction mode and style. In recent work [Chen et al., 2001] we have successfully linked the CoolTown web presence manager with our meeting agent system - using CoolTown sensors exposed via the web presence manager to detect changes in context. Those changes are then combined with knowledge of scheduled meetings to detect when planned meetings are actually occurring and to recommend relevant documents based on context. We will continue investigating ways of integrating CoolTown technology and models into our work. The primary difference is that in addition to the largely physical context and location-awareness exploited by CoolTown [Krishnan, 2000], CoolAgent adds more "semantic context", to represent intention, goals, tasks, etc.

5 The CoolAgent Partnership

We have been engaged in several cycles of collaborative testbed/solutions building to experiment with agent-based solutions construction and to evaluate several agent and web service technologies. We are exploring the rapid development of a prototype of a realistic multi-agent based system.

During Jan 2001 through June 2001 we executed a high-intensity collaboration, the so called "CoolAgent partnership". The goals of the partnership were to:

- Evaluate and demonstrate the concepts and capabilities of a variety of assistant and agent configurations
- Learn how to work across multiple organizations to rapidly create a medium sized agent-based solution for mobile professionals
- Create a testbed to try different models of agent intelligence and ease-of-use technology.
- Calibrate ourselves on what it takes to build such a solution using agents and current technology, and evaluate needs for better technology
- Experiment with algorithms for meeting scheduling and interaction with personal preferences
- Understand how this technology mix could scale to larger teams. Initially, we will deploy a small usability testbed as a team pilot on ourselves ("next-bench" or "eat your own dogfood" experiment)
- If robust enough, the prototype will be transferred to the other partners, who envision expanding, deploying, productizing and showcasing it in various ways

5.1 Organization

The CoolAgent partnership was a 23 person cross-organizational, geographically distributed team. People worked about 25%-75% time on this joint effort. Team members came from 5 different organizations within HP, and were situated in Palo Alto, Mountain View, Cupertino, Hawaii, and Colorado. Travel was restricted.

The project was divided into several 3-5 person cross-organizational teams, so that each organizational group was represented on each team. These teams were:

- Management/steering (scheduling decisions, final say on resource allocation, goal alignment)
- Scenarios/requirements
- Architecture and technology assessment
- Appliances (mostly UI, speech, natural language and customization)
- Agents (JADE, behaviors, rules, fuzzy logic)
  - Services (wrappers and interfaces to existing systems, like Microsoft Exchange)
  - Software engineering (processes, nightly build, multi-site code shipping, testing, conventions, ..)

5.2 Process

We used a fast-paced incremental/iterative approach, with an automated nightly build and Microsoft's VSS as source code control. We used automated FTP and Web servers to deliver nightly builds to partner machines. We allocated 8 two-week cycles for design, implementation and initial testing of features, and a few weeks for consolidation and buffer.

We prioritized features based on our initial assessment of how hard they were to do, how key each was to the concept, and how much research was needed to invent solutions.

Some of us wanted to use explicit use case models, some pair-programming (subset of XP) techniques and explicit refactoring steps, but the group was not ready to commit to that much process, even as an experiment. This was one of the tensions between research experimentation and creating potentially usable prototypes that we needed to keep in balance.

We did not do enough mechanized testing or invent or acquire enough testing tools. The predictable annoying feature integration problems were addressed in an ad hoc manner during the last week before the scheduled public demo.

Finally, we used and partially participated in the OpenSource process that provided us with several iterations of the JADE agent system. We submitted a number of enhancements to the JADE community that were adopted in later releases.

6 Results

6.1 The May/June 2001 Demo

The Phase 1 Personal Assistant demo showed a subset of a family of powerful productivity services that are combined by dynamic agent composition into intelligent solutions for the mobile professional. The demo and underlying technology was created by the CoolAgent partnership to show how intelligent agent technology, coupled with the web, Bluestone, speech, natural language, and a variety of real-world work-place services can be shaped and integrated. We have showcased this as a conceptual prototype of a flexible, intelligent, me-centric peer-to-peer workplace solution environment.

This demo was intended as a first step towards a deployable, usable prototype testbed for the "next-bench" test by the CoolAgent partners, and indeed was used in a limited way by several of them. As we will explain further, we learned that much more attention to robustness and security is needed, as well as additional features, before even motivated early adopters will be willing to commit their personal scheduling to such a system.

The essence of the demo is as shown in several of the Figures shown above.

Table 2 - highlights many of the Personal Assistant aspects we tried to show:

- Customization via personal information and context
- Multiple appliances, selection of appliance based on context
- Multiple, real-world services
- Interaction with other people and communities
- Agent negotiation and "rule-based intelligence"
- Natural communication via speech and natural language

There are several machines involved, each running one or more of the services shown in Figure 6. Our demo involved:

- 4-5 Individuals with Jornada and/or office desktop (NT or Linux) running browsers. Each user has 3 or 4 agents comprising their PA running on their behalf on a local machine.
- A central agent and server machine ("sandbox") running the JADE platform and the Bluestone totally-e-mobile system and many general service agents.
- Several other machines in Palo Alto, Cupertino, Mountain View and Atlanta were running a variety of real, existing services. Many of these services are the ones we usually use directly or via simpler tools in the course of a normal workday:
  - Speech <-> Natural Language
  - Natural Language <-> text
  - Exchange server interface
  - VoiceMail (Octel) interface
  - RoomBooker (HP Labs meeting rooms)
  - Reservit (non-HP labs meeting rooms)

- MeetingPlace (Teleconference phone number server)

Figure 6 and 8 show the use of Bluestone and JADE as a key piece of platform

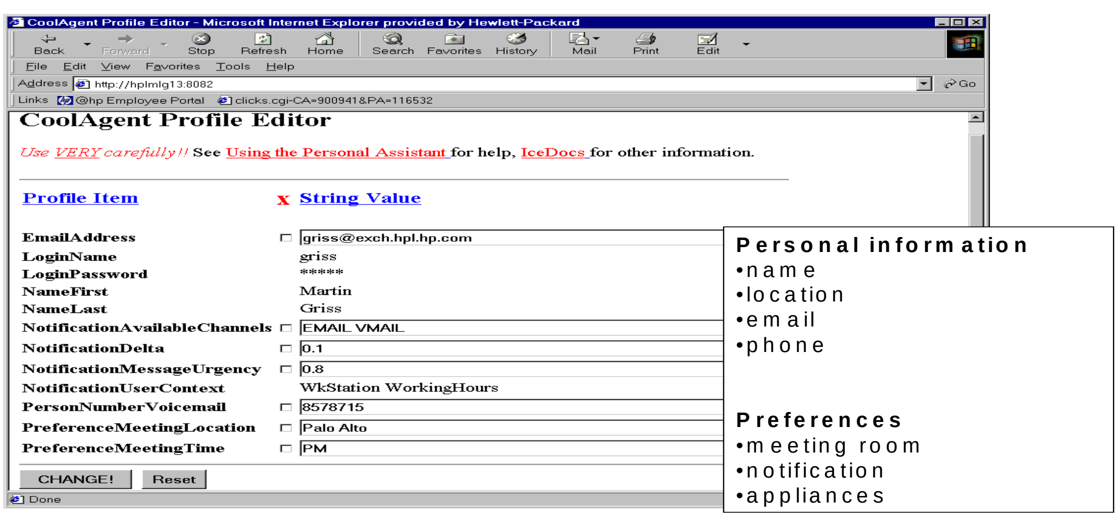
The primary demo scenario is:

1. Each member of our team has a Personal Assistant running somewhere, on their office desktop or the agent server.
2. Each Personal Assistant loads a set of preferences and personal data (consisting of an XML profile file, some properties and some Jess™ rules), and has access to or creates one or more local support agents.
3. As illustrated in Figure 10, one individual requests a meeting with one or more people, and/or with a named group. This request can be via a spoken command to a Jornada, a Web form from Jornada or desktop browser, or email English request.

   e.g. "*Setup a meeting with the Martin and Reed next Friday afternoon.*"
4. As shown in Figure 12, this request is converted into an Agent Communication Language (ACL) message and dispatched to the user's Personal Assistant, which then requests a meeting assistant to negotiate with the Personal Assistants of the intended participants, and/or team assistants for existing formal teams. The various Personal Assistants consult their owner's calendars and preferences, and attempt to find a ranked list of acceptable times.
5. Depending on the combined preferences, schedule and location, one or more conference rooms of appropriate size and features, and possibly a teleconference line, will be requested and reserved if available.
6. If all goes well, the meeting will be established, entries made in calendars if preferences permit, and notification to each participant will be sent via one or more preferred modalities, such as email, voicemail, pagers, or cell phones, using heuristics based on urgency, location, time, as illustrated in Figure 11.

Each agent (or the container housing several agents) can optionally host a light-weight HTTP server [Cowan and Griss, 2001], which can be configured to provide a CoolTown-like web presence, and also to view and optionally modify some of the agent and profile parameters. This also supports web-based editing of the starting profile (as shown in Figure 15).

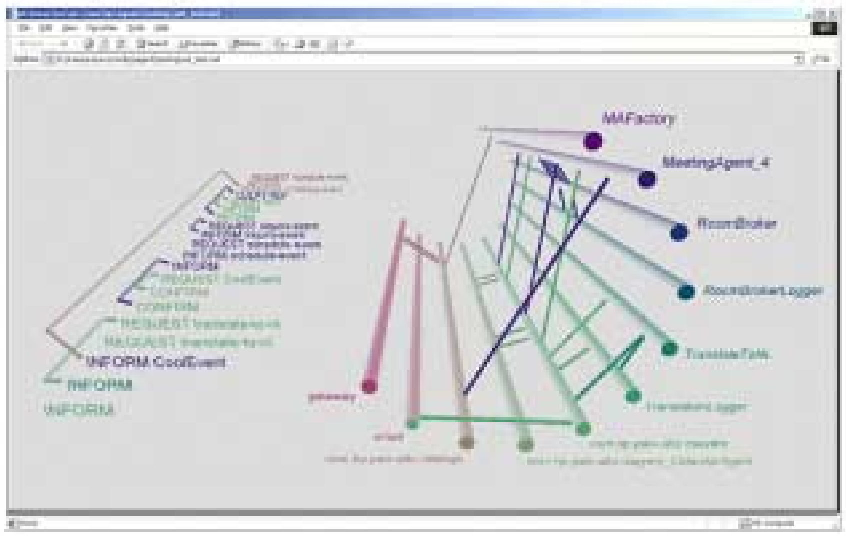
Figure 15: Web-based Profile Viewer/Editor



We also provide web interfaces to analyze the agent system. This web pages allow remote users to see which agents are active, to examine log files from running agents, and to view animations obtained by post-processing the log files. An example of such an animation is shown in Figure 16.

Figure 16 Web-based animation (using the Virtual Reality Modeling Language)

At the left is the sequence of messages from oldest (in the distance) to newest (closest). The links between messages show responses to earlier requests. At the right is the set of agents and the messages between them; again the most recent messages are closest to the viewer. During the animation the messages recede as new messages arrive.



6.2 The experimental testbed and user trial

Following the series of demos in May 2001, we felt the demo system had stabilized enough to increase the size and duration of our tests in the form of a limited user trial. We released the system to subsets of our 10-15 person self-use "next-bench" or so-called "eat your own dogfood" test team. The results have helped us prioritize key research and usability concerns.

We used our own distributed team as the first "live" customer base. We collected requirements from our experience and that of our admin assistants in setting up distributed meetings. We were also vision-driven in selecting and prioritizing requirements, to help provide a base for future systems with more features and richer domain. We are a good source of realistic requirements - we scheduled LOTS of distributed meetings.

We initially planned to do this "dogfood" test on the testbed for several weeks in May; it slipped and became a shorter test of a week during June. We have used the process and experience to refine and evaluate new features and improve performance of the system. Deliverables include design documents, overview and detail slides, the code, and the models and scripts to create the system and execute the testbed/sandbox, etc.

6.3 Key learnings

With some work on additional meeting specific and communication features (such as canceling and rescheduling meetings, use of the telephone), and some more richness in the handling of alternatives and over-constrained meeting situations, the sample application looks like it could readily and very usefully support a reasonable sized workgroup.

We also learned that we needed to improve the robustness of both individual agents and the system as a whole, and provide better tools to build these more robust agents and components. By robustness, we mean both the detecting and reasonable handling of a variety of exceptional conditions by agents, and appropriate support in the infrastructure for restarting agents and services.

Some of the exceptions relate to over-constrained meeting requests (e.g., no rooms available, or no compatible meeting times available), which can often be resolved by prioritized tradeoffs, exploration of alternatives, or dialog with the user. Other exceptions relate to system issues, such as dealing with services that are inaccessible, down or unresponsive. These can be dealt with by timeout/retry, trying alternative services, restarting services, watchdog monitors, etc.

Finally, we learned that security was a major issue, particularly the handling of passwords. Agents need to know the user's passwords if they are to access protected services on the user's behalf. Storing and transmitting passwords is a problem. While users are willing to type passwords to access calendars and roombookers, they are less comfortable delegating this authority to untrusted "experimental" agents.

7 Summary

With the May 2001 demo and subsequent limited "dogfood" test, we reached an important checkpoint in Aug 2001: a working testbed, comprising multiple agents, appliances and useful services.

We have built a medium-scale testbed. It makes use of several real-world web-based services, providing a flexible integration of capability. It already shows more promise than just a concept demo. This has provided a good base for demonstration, for testing new agents and tools, and to begin to attract new collaborators.

However, we also learned that several improvements were needed before the toolkit, testbed and sample application could be more widely deployed. Some of the next steps we envision and have started include: expanding the robustness of the system, using BlueJade (see also our work on Agentcities [Cowan et al., 2001; Cowan et al., 2002]); and increasing the power and flexibility of the meeting handling capabilities; adding some new email handling capabilities; and exploring some machine learning, and rule-based recommender agents [Katz, 2002].

ACKNOWLEDGMENTS

Our thanks to the numerous people who built, tested and demonstrated innumerable versions including, past and present colleagues, summer students, University collaborators and consultants, particularly: David Bell, Lew Creary, Harry Chen, Chris Davidson, Nathan Dykman, Steven Fonseca, Jimmy Hong, Ed Katz, Brian Kirkland, Jay Knitter, Debbie Lienhart, Mark Sabiers, Shadi Saifan, Craig Sayers, Sebastien Siva, Sovrin Tolia, Kim Venass, Delal Williamson, Farrell Wymore and Karen Xu.

REFERENCES

[Bellifemine et al., 1999] Fabio Bellifemine, Agostino Poggi and Giovanni Rimassi, "JADE: A FIPA-Compliant agent framework", Proc. Practical Applications of Intelligent Agents and Multi-Agents, April 1999, pg 97-108 (*See http://sharon.cselt.it/projects/JADE for latest information*)

[Bluestone, 2001] HP Bluestone Application Server (HP-AS), part of HP Netaction middleware suite, http://www.bluestone.com

[Chavez and Maes.,1996] A. Chavez and P. Maes. "Kasbah: An Agent Marketplace for Buying and Selling Goods". *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96).* London, UK, April 1996.

[Chen et al., 2000] Q. Chen, M. Hsu, U. Dayal, and M. Griss, "Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation," in *Proc. Autonomous Agents 2000*, June, Barcelona.

[Chen et al., 2001] Harry Chen, Sovrin Tolia, Craig Sayers, Tim Finin, Anupam Joshi, "Creating Context-Aware Software Agents," 1st Workshop on Radical Agent Concepts, NASA Goddard Space Flight Center, Greenbelt, MD, USA, 26-28 September 2001.

[Cowan and Griss, 2001] Dick Cowan and Martin Griss, "Web based Object Inspector," HPL Tech Report, HPL-2001-297, Dec 2001.

[Cowan et al, 2001] Dick Cowan, Martin Griss and Bernard Burg, "BlueJADE – A service for managing software agents," Hewlett-Packard Laboratories, Technical Report, HPL-2001-296, November 2001. Also, Proceedings of Workshop on Challenges in Open Agent Environments, AAMAS, Bologna, Italy, July, 2002. (To appear.)

[Cowan et al., 2002] Dick Cowan, Martin Griss, Robert Kessler, Brian Remick, and Bernard Burg, "A Robust Environment for Agent Deployment," Proceedings of Workshop on Challenges in Open Agent Environments, AAMAS, Bologna, Italy, July, 2002. (To appear.)

[Dawson et al., 1998] F. Dawson, et al, iCalendar Message-based Interoperability Protocol – iMIP, RFC 2445, November 1998

[Finin et al., 1997] T. Finin, Y. Labrou and J. Mayfield, "KQML as an Agent Communication Language," in Software Agents, J.M.Bradshaw (ed), MIT Press, Cambridge, Mass., p291-316, 1997.

[Fonseca et al, 2001] Steven P. Fonseca, Martin L. Griss, Reed Letsinger, Evaluation of the Zeus MAS Framework, Hewlett-Packard Laboratories, Technical Report, HPL-2001-154, July 2001.

[Fonseca et al., 2001a] Steven P. Fonseca, Martin L. Griss, Reed Letsinger, "An Agent Mediated E-Commerce Environment for the Mobile Shopper", Hewlett-Packard Laboratories, Technical Report, HPL-2001-157, July 2001.

[Griss, 2000] M.L. Griss, "My Agent Will Call Your Agent," *Software Development Magazine*, February. (See also "My Agent Will Call Your Agent … But Will It Respond?" Hewlett-Packard Laboratories, *Technical Report, TR-HPL- 1999-159*).

[Griss and Letsinger, 2001] Games at Work - Agent-Mediated E-Commerce Simulation, Workshop, Autonomous Agents 2000, Barcelona, Spain, June 2000.

[Guttman et al, 1998] R. Guttman, A. Moukas, and P. Maes. "Agent-mediated Electronic Commerce: A Survey." *Knowledge Engineering Review*, June 1998.

[Hayes Roth 2001] R. Hayes Roth, Personal Communication.

[Katz, 2001] Ed Katz, "A Multiple Rule Engine-Based Agent Control Architecture", *IEEE 6th International Conference on Intelligent Engineering Systems*, May 2002.

[Kindberg et al., 2000] Tim. Kindberg, et. al., "People, Places, Things: Web Presence for the Real World", *HP Laboratories, Technical Report,* HPL-2000-16, Feb 2000. (See also http://www.cooltown.hp.com).

[Krishnan, 2000] V. Krishnan, "Location Awareness in HP's CoolTown," Position paper, W3C/WAP Workshop on Position Dependent Information, Sophia Antipolis cedex, France, 15-16 February, 2000.

[Maes, 1994] Pattie Maes. "Agents that Reduce Work and Information Overload." Communications of the ACM, Vol. 37, No.7,pp. 31-40, 146, ACM Press, July 1994.

[Maes et al., 1999] Pattie Maes, R.H. Guttman, and A.G. Moukas. "Agents that buy and sell." Communications of the ACM, Vol.42, No.3, March 1999, pp.81-91.

[Nwana et al., 1999] H. Nwana, D. Nduma, L. Lee, J. Collis, "ZEUS: a toolkit for building distributed multi-agent systems", in *Artificial Intelligence Journal*, Vol. 13, No. 1, 1999, pp. 129-186. (See http://www.labs.bt.com/projects/agents/zeus/).

[O'Brien and Nicol, 1998] P.D. O'Brien and R. Nicol, FIPA: Towards a standard for intelligent agents. BT Technical Journal, 16(3), 1998. Also see the Foundation for Intelligent Physical Agents (FIPA) website, http://www.fipa.org

[Sayers and Letsinger, 2001] Craig Sayers and Reed Letsinger, "An ontology for publishing and scheduling events and the lessons learned in developing it", HP Labs Technical Report, HPL-2002-162, June 3, 2002.