MP 06W0000080

MITRE PRODUCT

Pervasive and Ubiquitous Computing in a Corporate Conference Room

September 2004

Lisa D. Harper
Abigail S. Gertner
Paul M. Herceg
Thomas T. Hines
Edwin H. Kemon
David E. Mireles
Michael E. Shadid
James A. Van Guilder

MITRE
Corporate Headquarters
McLean, Virginia

Pervasive and Ubiquitous Computing in a Corporate Conference Room

L. Harper, A. Gertner, P. Herceg, T. Hines, E. Kemon, D. E. Mireles, M. Shadid, J. Van Guilder

The MITRE Corporation

7515 Colshire Drive

McLean, VA 22102-7508

+1 703-883-5241

lisah@mitre.org

Abstract

This paper discusses pervasive and ubiquitous computing in a smart room environment. The system we describe resides in an operational conference room at The MITRE Corporation in McLean, Virginia. This environment is currently used by a small number of volunteer participants in order for us to accommodate user input early and frequently in the development cycle. In this environment, an embodied agent assists meeting participants with operating room devices, including those used for video teleconferencing (VTC). Primary system components include an interaction manager, avatar, speech recognizer, text-to-speech engine, room control application, macro player, biogate (face recognition) system, and context model. We are also introducing location-tracking and vision-based gesture recognition in the near future.

1 Introduction

A ubiquitous computing smart room environment is composed of both *infrastructure systems* that are associated with a given locale and *personal systems* that are mobile or wearable [WANT2002]. We have developed an operational, pervasive smart room by enhancing a corporate videoconferencing room, or *Team Room*. Team Rooms at MITRE are more than just meeting rooms; they are state-of-the-art audio-video (AV) conferencing rooms which support local and distributed meetings. These rooms offer meeting participants the convenience of controlling many devices from a single interface. Devices in our environment include two plasma screens, multi-point VTC, telephone, room audio, VTC camera, satellite television, VCR, lights, window shades, and more. Our agent controller also has access to room PCs and can interact with PowerPoint briefings and other common applications. Team Rooms are complex environments with a large array of capabilities.

Because Team Room environments are so complex, we've observed that there is often much time lost in meetings during room setup. Users spend a lot of time pushing buttons on the main room control panel trying to put up presentations and make remote connections. The goal of our investigative work is to reduce the task load for the user and replace the current control panel with a more intuitive, context-aware interface. Users interact with our agent using speech or GUI. Our agent controls room devices and room PCs. It executes actions based on a task model and uses sensory input in building a model of context. A dialog manager, or interaction manager, tracks the human-to-agent conversation. Though the system is functional, we are still gathering information from users via Wizard-of-Oz studies intended to help us develop and expand our grammars and task model.

This paper presents the design of our intelligent room system. It is intended to augment the overview description and related work provided in [HARP2004].

2 Smart Rooms and Ubiquitous Computing

Lyytinen's clarification of three aspects to ubiquitous computing is particularly helpful to the discussion of smart rooms [LYYT2002]: *mobile computing* increases ones ability carry computing services independent of location; *pervasive computing* enables personal devices to use the environment to dynamically build models of computing; and *ubiquitous computing* enables a personal device to compile and re-use information that transcends multiple pervasive environments.

In a smart room, *pervasive* infrastructure systems detect *pervasive* personal systems and offer services. The handshaking of personal systems with the room's infrastructure systems initiates numerous activities including updating context, querying profiles and privileges, dynamically building user interfaces, and broadcasting events to subscriber processes. Service to the personal system may include distributing processing to the more capable infrastructure systems. Depending on the application domain of the smart room, a personal system may not be required. In the absence of such a device, some method of authentication and authorization may be desirable.

To support personal systems (e.g., PDAs, microphones, location tracking badges, and sensors), the system needs some sort of service registration and service discovery framework. Several architectures and protocols have been proposed by researchers and consortiums and are out of the scope of this paper. However, worth mentioning are the following: Microsoft's Universal Plug and Play (UPnP), the Salutation Architecture, Home Audio Video Interoperability (HAVi), Jini (Sun's version of plug and play), Service Location Protocol (SLP), and Open Service Gateway Interface (OSGi) [FRID2001]. Others include HP Cooltown, and JADE LEAP.

A smart room relies heavily on a context model for maintaining data about changes in the environment [HATT2003], [CHEN2004]. The model collects data about the changing state of infrastructure systems and leverages additional information that can be gained from the network outside the immediate environment (e.g., MS exchange servers, intranet, internet). It fuses information about activity in the environment, draws inferences about activity in the environment, and reports events to interested subscriber systems. The context model itself relies on device events, sensors, cameras, RF or UWB receivers, and the like.

It is important to mention that smart rooms may also support remote access, although this is considered outside the limits of the ubiquitous computing environment. A voice or video call into the room can enable remote participants. *Remote systems* can be thought of as a container capable of downloading and hosting an application with an interface to the smart room. In this case, the service must take into account the capabilities of the remote system (e.g., a PDA system would have much less capability than an automobile hosted system). If employed appropriately, the downloaded application would exploit an automobile system's surround sound, camera, and heads-up display to enhance the near-presence of the remote participant.

3 Motivation

Our organization hosts a network of state-of-the-art VTC rooms. Room equipment typically includes a pan-tilt-zoom camera (or PTZ), microphones and speakers, an audio mixer/switch, a video switch, a codec hosting an H.323 endpoint, multiple displays/projectors, motorized projection screens, a satellite television receiver, a VCR, a lighting control system, motorized window shades, and more. Multipoint meetings are switched via a multi-function controller (MFC) video bridge in the network. An ISDN service enables videoconference connections with outside organizations. The analog telephone line is used for a connecting a single party or for connecting multiple participants via an audio bridge.

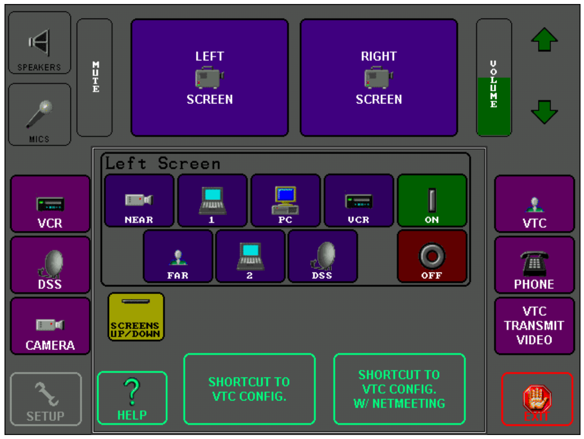

Figure 1. Complex AV Controller Interface

The *AV Controller* centrally controls all standard Team Room devices including VTC and telephony connections, plasma screens, and camera, overhead lights and window shade settings. Essentially, it is a proxy because it acts on behalf of other devices in servicing requests. The *AV Controller Interface* – the user interface (see Figure 1) – is a touch-panel screen with a collection of button-panels, which are tailored to specific devices and functions. Over time, the number of Team Room devices and features at MITRE has expanded, increasing the complexity and quantity of the panels. It is impractical to train all employees on this user interface – in actuality, panels vary slightly from room to room since not all Team Room configurations are identical. However, the lack of user training and sporadic frequency of use can account for meeting delays of up to 10 to 20 minutes. We hypothesize that a human avatar could reduce meeting delays, particularly at the start of meetings, thereby increasing participant productivity.

We also hypothesize that by better integrating sensors, multimodal devices and intelligent software, perceptive assistive agents can reduce the cognitive burden on humans allowing them to focus their attention on decision-making and social interaction necessary to achieve complex team tasks. To test this we constructed an experimental AV facility with user-observation capabilities – the *Experimental Team Room (ETR)* – and we developed an embodied assistive agent with an integrated network of sensors - the *Electronic MITRE Meeting Assistant (EMMA)*.

4 The Agent Controller

In the ETR, meeting participants collaborate with EMMA to setup the room and make changes in room configuration. We created a dialogue manager using the Mitsubishi Electronic Research Laboratories (MERL) Collagen system. The dialogue manager is responsible for understanding and generating conversation. It relies on discourse plans and a task model to do this. Rather than reiterating the Collagen design and approach in this paper, we refer the reader to [RICH2001] and [RICH2002]. We also use the terminology found in these works, such as collaboration, collaborative discourse, recipe, discourse state plan tree, etc.

In the ETR, participants have the following options: controlling devices manually, controlling them via a GUI panel, or controlling them in a mixed-initiative interaction with EMMA. We believe that many users will prefer a mixed-initiative interaction where they take some actions themselves but also delegate some actions to EMMA.

Currently, EMMA intervenes only at the request of a user. EMMA is essentially modeled on a user-initiated interface. We have also experimented with a tutorial-based system-initiated approach and feel that eventually both conversational styles may be useful depending on the knowledge and experience of a particular user in performing particular tasks.

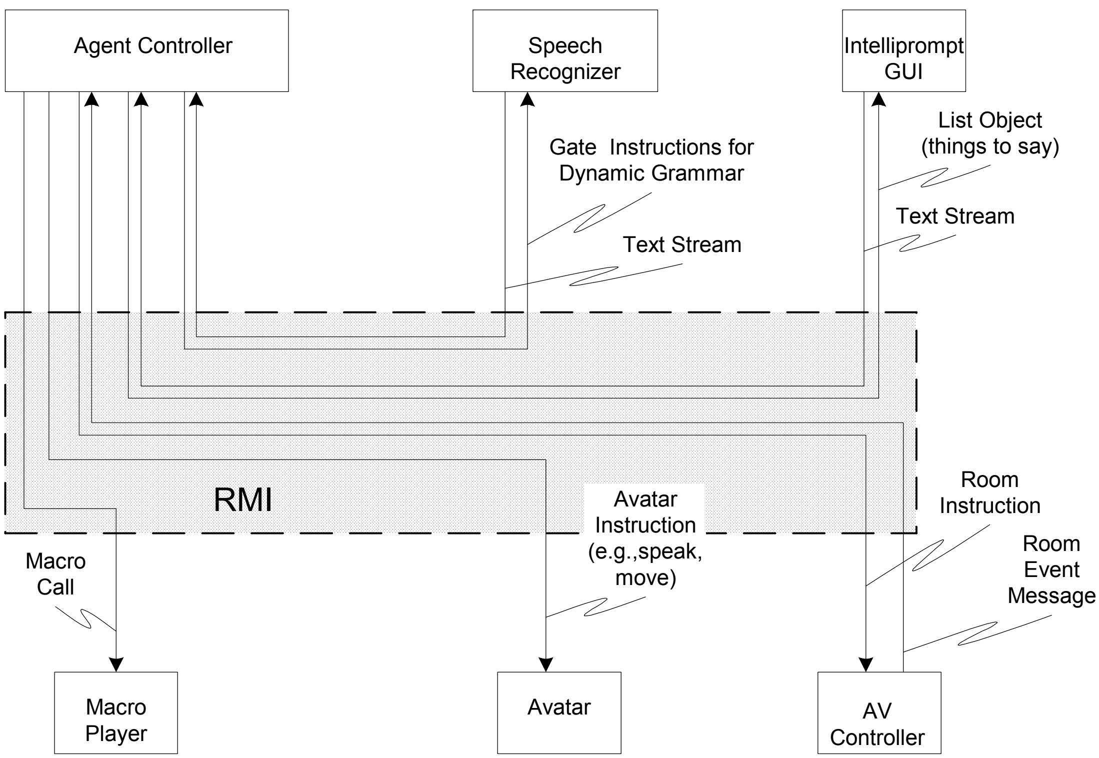

Figure 2. Distributed EMMA system.

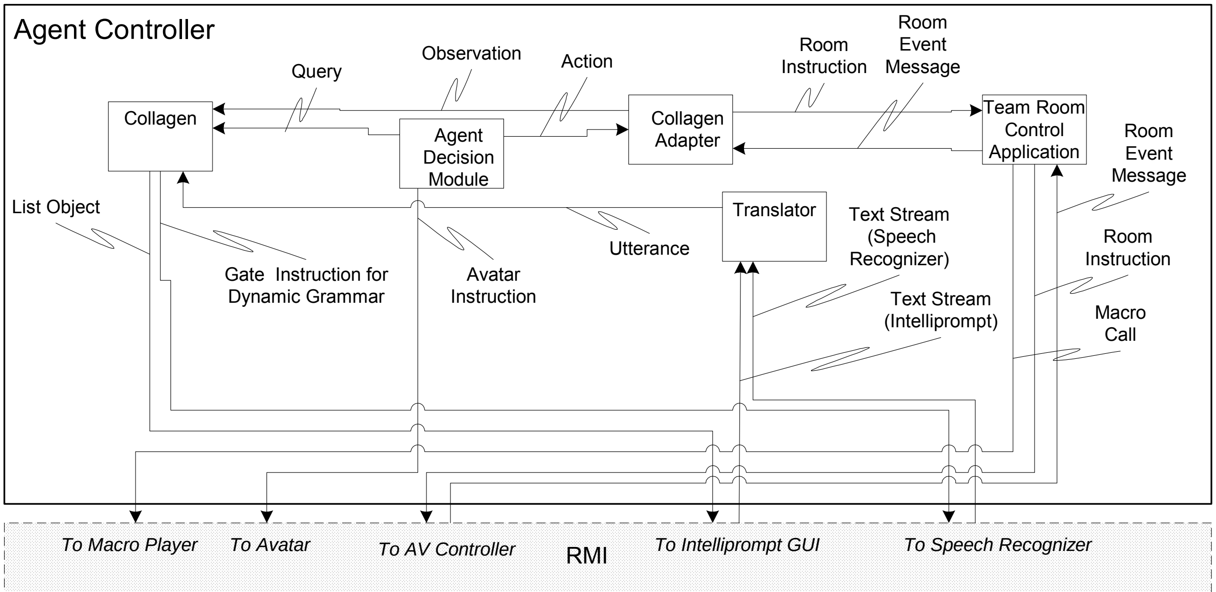

Figure 3. Agent Controller design.

Figure 2 shows the Agent Controller in the context of the entire EMMA system. Figure 3 shows a decomposition of the Agent Controller. EMMA's task model consists of rules for achieving tasks in the domain. These rules, or recipes, are used to dynamically construct the discourse state plan tree as the discourse unfolds. At startup *Collagen* generates an initial plan tree and the resulting agent agenda and user agenda. It then sends a grammar and list objects to the Speech Recognizer and "IntelliPrompt" GUI, which presents the user agenda as a list of valid utterances or actions.

When the user speaks or indicates a valid utterance, Collagen receives via the *Translator* an utterance object. When the user operates a device directly, Collagen receives via the *Collagen Adapter* a corresponding observation object. Collagen determines how observations, utterances, and recipes contribute to goals in the current discourse state plan tree or to a new goal. It updates the discourse state plan tree, agent agenda, and user agenda, accordingly. Grammars and list objects are also updated, accordingly.

The *Agent Decision Module* detects the Collagen update, then queries the agent agenda and selects a domain action (e.g., connect a VTC, show a PowerPoint briefing, dim the lights) or a discourse action (e.g., speak a proposal, move), as appropriate. Collagen monitors this agent behavior in the same manner as the user's behavior (i.e., with utterances and observations), and proceeds with the corresponding updates. This interaction continues – with the agent and user taking turns.

The *Team Room Application* enables the Agent Decision Module to perform actions to control the room, both using the AV Controller and a Windows-based *Macro Player* that we developed as a part of this system. The Team Room Application also receives room event messages reporting any changes to the AV environment[1]. The *Collagen Adapter* converts room event messages to observation objects for consumption by Collagen.

The architecture we are using is configurable; we can replace the Collagen dialog manager with an application that lets a human masquerade as the agent. In this manner we have performed several user evaluations known as "Wizard-of-Oz" (WOZ) experiments. These experiments are designed to help us understand more about how users naturally desire to interact with our system. Experiment subjects speak to EMMA using a hold-to-talk button and are given only brief instructions on the kind of language that they should use. We convey that they should speak in short utterances and that her speech recognition is limited to a relatively small domain. A human wizard does the actual language understanding and response generation (through the EMMA avatar) but uses the EMMA system for carrying out actions. These experiments have been useful in informing our system design and also for testing and bug fixes.

The Agent Decision Module uses the task model reflected in the recipe library as its domain knowledge. It uses it to generate and interpret observations. Initial user evaluations have revealed patterns in task performance that span across users familiar with Team Room operation. Designing the task model has been an iterative process of developing recipes and testing system behavior in interaction with users.

5 The Macro Player

EMMA controls the ETR's Windows PC in addition to other Team Room devices. As with the AV Controller, a Windows PC is shared between meeting participants and EMMA[2]. Therefore we developed a Windows-based *Macro Player* that could take parameters; for example, executable names or user names[3]. We also decided that non-programmers would need to write the macro scripts, so a

---

[1] Participants can use the AV Controller Interface to monitor changes to the environment and to perform actions simultaneous with EMMA.

[2] This PC hosts typical desktop applications, including a collaborative white board.

[3] The commercial products we reviewed did not support such parameters.

corresponding scripting language was developed. The macro scripting language is simple, consisting of primitive *actions* and *conditions,* which can be combined into *sequences*. At runtime, the macro player parses the macro scripts to instantiate macro objects that are available to the Team Room Application. As a result, the Macro Player enables EMMA to bring up PowerPoint presentations, conduct web searches[4], or other novel actions developed by the scripter.

6 User Interface

Interaction with EMMA is multimodal – via speech (i.e., audio) or a GUI (i.e., point and click). We are also in the process of extending her ability to recognize a small set of arm gestures as well as location relative to room zones or devices. At each point in the discourse, there is a list of appropriate phrases, or utterances, according to the current discourse state plan tree. Collagen sends instructions to the speech recognizer to enable and disable parts of the grammar; this tailors the grammar to recognize only phrases appropriate for the current user agenda. Simultaneously, it sends a list of the same phrases to the IntelliPrompt GUI (web-based interface currently under development). In addition to displaying user utterances and actions, the IntelliPrompt GUI has an embedded push-to-talk button and display space for our half-body avatar.

Figure 4. EMMA

7 Inter-Process Communication

EMMA is primarily written in Java. Java Native Interface wrappers are used where components are written in C. Therefore, we used Remote Method Invocation (RMI) to distribute the EMMA system across several platforms in the form of clients and servers. Planned for the next release, a common application class was developed to standardize the way each module plugs into the distributed system.

4 Participants have the option of using the applications directly.

The same class provides object management functions: registration, start, stop, enable, disable, etc. These enhancements establish a framework for extending and maintaining the EMMA system. This framework will also be used to host pervasive personal systems.

8 The Biogate

We developed the *biogate* as a place for participants to register into the room and pick up devices to be used during the meeting (e.g., a microphone, a location tracking badge). Upon entry to the front room participants register by undergoing face recognition. The biogate displays the planned and present participants. Once integrated with the larger system, this component will report to EMMA the device-to-user associations for the purpose of inference. An overhead camera enables remote observation and recording.

As with most face recognition systems, the users must undergo an enrollment process prior to recognition. The commercial system in the ETR captures approximately 100 face images during enrollment. Enrolling multiple times increases the number of images and recognition accuracy.

9 Facility design

The ETR facility was constructed with both a front room meeting area, and a back room for equipment and observation (see Figure 6). Two panes of 3' x 6' one-way glass were installed to separate the rooms and enable meetings to be observed from the back room. A three-inch raised floor was added for cabling between the rooms. Air conditioning and high density power (i.e., several 20 amp circuits) were configured for generator backup. Electrical outlets and cable trays were installed above the drop ceiling in anticipation of camera and projector equipment. And standard six-connector network jacks were installed every 4 feet.

In the front room, users view four systems: the VTC connections, the PC with the EMMA avatar and IntelliPrompt GUI, the Windows PC for desktop applications, and the AV Controller Interface. Wireless keyboards and mice are used as needed for the desktop applications and IntelliPrompt GUI. The PCs and AV equipment rack reside in the back room.

Since the facility and equipment could not be purchased with limited research funding, it was constructed as a corporate resource with the agreement that it would be available for employees' use twice per week. Although seemingly incompatible, we manage both operational and development systems and provide a switching mechanism between the two. We will be configuring the wireless keyboards/mice, microphones, and AV system to be switched bi-weekly between EMMA test systems and EMMA development systems. Currently, users are limited to using our development system.

10 Context Model

One of the challenges in developing smart room technology is providing an environment that will respond appropriately and timely to users' needs. Prior research has recognized that intelligent environments can benefit by using models to maintain and evaluate the context in which actions in the environment occur [HATT2003],[JUDD2003], [PETE2003]. As part of our continuing work we are integrating a model to maintain contextual information about meetings, participants, and conference rooms. EMMA and devices in the environment will make use of this model to provide more context-appropriate responses to users' requests and actions. As similarly shown in other research [MUHT2003], [CHEN2004] this model will also contain logic to maintain user preferences, authorizations and permissions.

The context model consists of three primary components: a knowledge-base, a reasoning engine, and an agent interface. The knowledge base was implemented in Protégé [PROT2000] as a database-backed OWL[BECH2004] ontology. The ontology characterizes the concepts and relations of objects specific to a meeting, users, and the environment (sensors and other room devices). We are currently testing reasoning engines that have been developed as plug-ins for Protégé, and are initially using the Algernon [HEWE] plug-in. The agent interface is managed using the Java Management Extensions (JMX) [SUN2002], which provides a flexible distributed access to the knowledge base and reasoning engine using any of a number of protocols such as, SOAP, HTTP, RMI, or CORBA. EMMA, infrastructure systems, and external calendaring systems will interface to the context model to maintain and access room state information as well as user and meeting specific information.

11 Location Tracking

Making EMMA aware of participant identity and location enables the execution of zone-based actions. As an individual enters the "presenter's" zone, EMMA can display the individual's profile to the meeting attendees. We've investigated various ad-hoc and infrastructure based system location tracking systems. The Ubisense ultra wideband (UWB) based solution was selected for our implementation. The Ubisense system uses a network of ceiling-mounted sensors in conjunction with a central server. The associated Ubitags, attached to individuals or assets, are in constant communication with the network of sensors.

Time of flight calculations are used to determine each Ubitag location. Since the system has an accuracy to within 5cm (95% of the time), we hope to also use it for gesture detection.

12 Component Launcher

Managing the various distributed systems in the ETR requires extensive user interaction. To launch the systems' distributed applications, the administrator must start processes on a number of different computers and in a specified sequence.

Mother of All Launchers (MOAL) was developed to allow the administrator to interact with a single computer to configure and launch applications across multiple computers. All software setup and configuration is managed by the MOAL in an *action plan* to enable rapid execution (or reproduction) of a particular distributed application. By automating the launching of distributed applications, human error is avoided and setup time is significantly reduced. This improves configuration management via a remote management interface, which gives the administrator centralized access to each of the application consoles (see Figure 7).

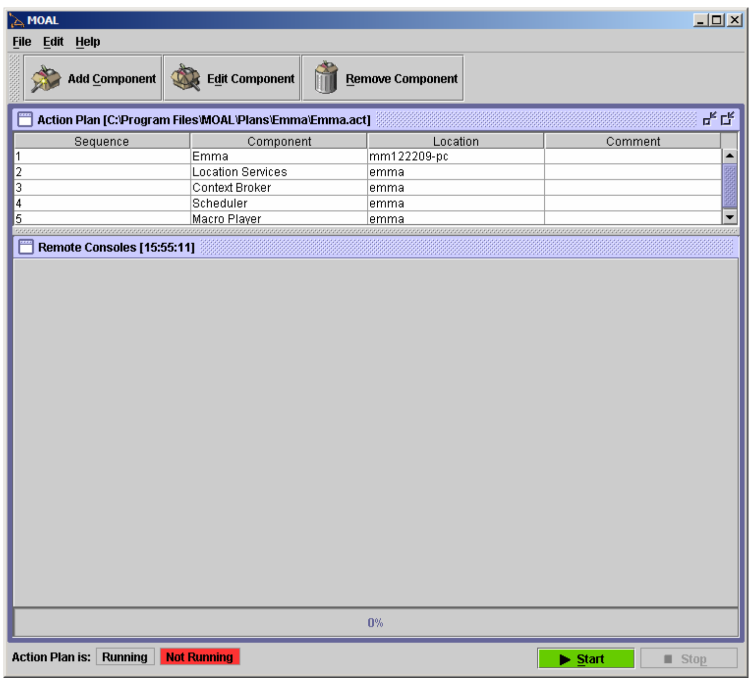

Figure 6. An action plan in MOAL.

13 Conclusions and Next Steps

In this paper we have presented a ubiquitous and pervasive computing system that integrates a perceptive assistive agent with an operational VTC room. At the writing of this paper, the agent only perceives user utterances and events via the AV Controller. This year, we are working toward increasing sensor information in order to make EMMA more context-aware (e.g., location tracking, integrating the biogate, user profiles/preferences, Exchange calendars, etc.).

Plans include adding the UbiSense ultra-wideband (UWB) location tracking system to locate people and room objects in real-time. UbiSense badges will be assigned at the biogate. In addition, we will extend the AV Controller to monitor microphone audio levels and assign lavaliere microphones to participants as they enter at the biogate. We will then extend EMMA to correlate participant audio level, audio duration, and participant location, in order to select the best camera preset for remote VTC participants (similar to the intelligent cameraman in [SHIY2003]). We are also integrating EMMA with the corporate scheduling system, which will enable her to anticipate meetings and have access to presentation files to load into PowerPoint. These and others are opportunities to have EMMA correlate information similar to [CHEN2004]. Supporting user profiles/preferences in conjunction with the biogate will enable dialog with EMMA to be more personalized (e.g., EMMA will substitute the user's name in the phrase "bring up *my* intranet folder").

The ETR system was initially built by collecting old VTC equipment and PCs, so the design was not necessarily scalable. Modifications are planned to make the system and software more scalable and extensible. This will enable the room to host concurrent and follow-on HCI research projects. For example, in future work we hope to introduce speaker identification.

This system was created to test the hypothesis that a human-like agent can streamline room operation and increase meeting productivity. Therefore we plan to conduct usability comparisons between the AV Controller Interface and EMMA using MITRE volunteers.

14 References

[HANS2002] Hanssens, N., Kulkarni, A., Tuchinda, R. and Horton, T. "Building Agent-Based Intelligent Workspaces". In *ABA Conference Proceedings.* June 2002.

[HARP2004] Harper, L., Gertner, A., Van Guilder, J. "Perceptive Assistive Agents in Team Spaces", In *IUI'04*, Jan. 13-16, 2004.

[SHIY2003] Shi, Yuanchun, Xie, W., Xu, G., Shi, R. Chen, E., Mao, Y., Liu, F. "The Smart Classroom: Merging Technologies for Seamless Tele-Education". *IEEE Pervasive Computing*, pp. 47- 55, Vol. 2, Issue 2,April-June 2003.

[RICH2001] Rich, C., Sidner, C. L., Lesh, N. B., "COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction", Artificial Intelligence Magazine, Winter 2001 (Vol 22, Issue 4, pp. 15-25)

[RICH2002] Rich, C., Lesh, N., Rickel, J., Garland, A. "A Plug-in Architecture for Generating Collaborative Agent Responses", *International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, ISBN: 1-58113-480-0, pp. 782-789, July 2002 (ACM Press).

[ADDL2002] Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A. "Implementing a Sentient Computing System", *IEEE Computer*, August 2001.

[CHEN2004] Chen, H., Finin, T., Joshi, A. "A Context Broker for Building Smart Meeting Rooms" Proceedings, *Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium*, *2004 AAAI Spring Symposium*, March 2004.

[WANT2002] Want, R. et al. "Disappearing Hardware" IEEE Pervasive Computing 2002.

[LYYT2002] Lyytinen, K et al. "Issues and Challenges in Ubiquitous Computing", Communications of the ACM. December 2002.

[FRID2001] Friday, A. et al. "Supporting Service Discovery, Querying and Interaction in Ubiquitous Computing Environments." ACM 2001.

[HATT2003] Hattori, M., Cho, K., Ohsuga, A., and Isshiki, M. Context-Aware Agent Platform in Ubiquitous Environments and its Verification Tests. In 1st Annual IEEE International Conference on

Pervasive Computing and Proceedings of the 1st Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'03), March 2003.

[JUDD2003] Judd, G., and Steenkiste, P. Providing Contextual Information to Pervasive Computing Applications. In 1st Annual IEEE International Conference on Pervasive Computing and Proceedings of the 1st Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'03), March 2003.

[PETE2003] Peters, S., and Shrobe, H. Using semantic networks for knowledge representation in an intelligent environment. In 1st Annual IEEE International Conference on Pervasive Computing and Proceedings of the 1st Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'03), March 2003.

[MUHT2003] Al-Muhtadi, J., Ranganathan, A., Campbell, R., and Mickunas, M.D. Cerberus: A Context-Aware Security Scheme for Smart Spaces. In 1st Annual IEEE International Conference on Pervasive Computing and Proceedings of the 1st Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'03), March 2003.

[PROT2000] Protégé-2000. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/

[BECH2004] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A., OWL Web Ontology Language Reference, W3C Recommendation 10, February 2004.

[HEWE] Hewett, M. Algernon-J Rule-Based Programming. http://algernon-j.sourceforge.net/

[SUN2002] Sun Microsystems, Inc. Java™ Management Extensions Instrumentation and Agent Specification, v1.2, 2002.