The University of Kansas

Information and Telecommunication Technology Center

Technical Report

Ambient Computational Environments

Gary J. Minden and Arvin Agah

ITTC-FY2005-TR-20680-01

April 2005

Project Sponsor:
The National Science Foundation
Grant: EIA-9972843

Ambient Computational Environments

Final Report

1 Introduction

Ambient Computational Environments (ACE) is a project intended to explore the use of computers in our everyday environment. We addressed the questions of how to embed computation into our laboratory/office environment and how to organize, manage, and use such an embedded system.

The National Science Foundation provided significant funding through a Research Infrastructure grant to build ACE. The University of Kansas (KU) provided matching funds and DARPA and Sprint Corporation provided support for students and staff. The NSF grant supported research in the Electrical Engineering and Computer Science Department (EECS), The Information and Telecommunications Technology Center (ITTC), and other laboratories at KU.

This document reports the results of the ACE project. First, we describe the goals of the project. Most of the material following in Section 1 is taken from the original proposal. Section 2 describes our approach to the research and Section 3 describes the project structure. A year-by-year description of research accomplishments is provided in Section 4. Section 5 lists the major publications supported by the project and Section 6 concludes the report.

1.1 Architecture and Prototype of an Ambient Computational Environment

We proposed to research the design, development and initial implementation of an Ambient Computational Environment (ACE). The concept begins with the idea that computation resources, in the broadest sense, are readily available in our offices, conference rooms, auditoriums, and hallways. Second, users co-opt, with authorization, the computational resources within their proximate area. Third, users access computational services that are long-lived and extremely robust. And fourth, users interact in multiple ways with the Ambient Computational Environments.

We proposed to research the impact of ACEs on high-performance networking systems. The type of traffic in an ACE is likely to be quite different from conventional FTP, Web transfers, and large dataset access. Further, we proposed to investigate the mechanisms needed to secure transmission of ACE content over widely distributed next generation Internets.

ACEs represent a significant application and networking protocols driving Next Generation Internets. Networking Systems research has had a difficult time answering the question, "Where's the data?" Certainly, high-resolution imagery and video is one source. We believe that low-latency, highly reliable, interactive traffic, as typified by ACEs, is a major driver for emerging network protocols. We proposed to develop such traffic applications, the necessary network protocols, and measure network performance, and do so in the wide-area networking context.

While a comprehensive research program was proposed and a project initiated in June 2000; the project was de-scoped by the government in January 2002 and terminated in June 2002. During the limited active time of the project we were able to specify and design many of the software components for ACE and started some of the software development. This report provides an overview of the project, describes the documentation presented in a series of technical reports and the project management.

1.2 Ambient Computational Environments

The following vignettes illustrate our concepts. Herman completes editing his presentation in his office environment. He picks up a small lightweight device, we call a Personal Interactive Device (PID), and heads down the hall toward the conference room. The conference room is equipped with tabletop display screens, video/computer display projectors, sound system, microphone system, controllable video camera, and controllable lighting. Once there, Herman identifies himself through his organizer and requests his conventional working context be brought up on one of the table-top displays. He requests, and is granted, access to the conference room resources. From his working context, he arranges to bring his presentation to one of the data projectors, set lighting levels, arranging video cameras pointed toward the speaker position, directing a remote video feed to a second projector, and so forth as he sets up for his presentation. Herman has co-opted the conference room's resources to support his specific presentation. He knows he can do the same thing in any of the center's conference rooms or colleague's offices.

The last time Holly checked, her working context had been running for 497 days. Today, MIS is scheduled to swap her primary computer, display, and input station for a newer model. When the MIS support person arrives with the new equipment, Holly clears a path to her "computer", he quickly unplugs the display, keyboard, and computer modules to make room for the new equipment, disregarding the fact that the computer is running and, for the time being, is Holly's port to the computational utility. He plugs in the new equipment, turns it on, and leaves. Holly identifies herself to the new machine with her organizer and it quickly displays her working context. Tomorrow when she checks the "uptime", it will show 498 days.

A few weeks later, Herman has a second presentation. Only this time, when he enters, the conference room senses his entry and turns on the ambient light. Herman sits at a table top station and presses his thumb against a reader. His thumb print identifies him and brings up his working context on the nearby display. Herman directs, through gestures, voice commands, and conventional computer commands, to put the presentation on the right screen, point "that video camera" at "that seat", "put the remote video feed on the left screen", and so forth. The conference room reacts to Herman's voice commands, gestures, and computer mediated commands. Herman feels immersed in an Ambient Computational Environment.

The concepts of Ambient Computational Environments are the following:

Computational resources are readily available throughout the space in which people move. By "computational resources " we mean CPU cycles, memory, storage, display, wired and wireless communications, sound input and output, video input and output, i.e. anything connected with computing.

Users co-opt computational resources in their vicinity for their use.

Computational sessions are long-lived, and mobile beyond the extant of individual machines or instantiations.

The computational environment re-acts to user voice commands, gestures, and computer commands and maintains an individual model of how specific users act.

Our vision embeds low cost and high performance future computational units in our everyday environment. Offices will have computational resources, conference rooms will have computational resources, taxis will have computational resources, and airplanes will have computational resources, along with a multitude of other environments. Our vision distributes computation throughout our environment which can be co-opted for our use with proper authorization. Identification to the environment either permits or denies access to the local computational environment. One only need provide their valid identification to access available resources. This is contrary to the concept that we will carry around with us a "tool belt" of information appliances to address our needs. The authors of this proposal are tired are carrying ten pound computers, pagers, cell phones, and other personal devices everywhere they go.

The Ambient Computational Environment is different from conventional mobile computing or agent systems. In mobile computing, users lug their own computer to different locations and their computer maintains their working context. In other cases, users access a central location for their files and context depending on an underlying network communications infrastructure. In either case, the focus is on communications from a fixed resource, e.g. the laptop, back to a single central system. In agent systems, programs, called "agents," are launched into an interconnected set of computers to compute and carryout tasks for the launching entity. However, any movement of the user is ignored. Agents are disembodied from the actual user. Our vision is that users "carry" their working context with them either via mechanical means (a personal organizer) or individual identification and the computational environment is available and adapts to the individual's requirements and behavior.

Our vision is distinct from other recent concepts outlining future computer system organizations presented by Norman [1] and Kozyrakis and Patterson [2]. In these proposed concepts, the low cost and high performance of future computational units will be used to create ever more powerful mobile computers. Norman argues for a plethora of "information appliances." Kozyrakis and Patterson propose integrating the functions of calculators, personal organizers, multiple wireless phone services, paging, and audio/visual remote controls, and broadcast radio into a single personal unit. They further argue that multi-media applications and data streams will change the basic underlying computer architecture.

Sun's Jini™ architecture [3] considers "mechanisms for machines or programs to enter into a federation where each machine or program offers resources to other members of the federation." While Jini™ technology is one possible technology base for Ambient Computational Environments, it is only a network technology and does not describe the necessary services to build a robust environment, nor the necessary knowledge systems to provide intelligent reaction to user commands.

ACEs will have significant impact on network usage. Implementing persistent storage will require extremely low latency communications protocols. Interaction between ACE components is likely to be more of a transactional nature rather than today's client/server (GET/PUT) nature of the world wide web. Multiple high definition video and audio streams will require at least two orders of magnitude capacity over today's streaming video sessions. Understanding these shifts in network load and behavior is a major component of the proposed research.

1.3 Research Topics

We have identified seven areas of research necessary to build Ambient Computational Environments:

- System architecture of mobile programming contexts; access to contexts from multiple, remote locations; and persistent storage of contexts.
- Network protocols to support persistent storage systems over wide areas, transactional based communications, and highly mobile routing techniques and network behavior under ACE loads.
- Tools and techniques for access to and interaction with environments.
- Language and run-time systems to support long-lived, mobile computing contexts.
- Task negotiation among multiple user workspaces.
- Context-sensitive information retrieval.
- Disambiguation of spoken commands and gestures within different environments and multiple user contexts.

These research topics represent the significant range of activities necessary to build an ACE. Access to an ACE requires communication services, as well as efficient and intuitive user interfaces, which in turn require innovative interaction devices and methods. Environments that support long-term, mobile computations place new demands on the programming languages, underlying program support systems, and communication systems. Management of user intentions, history, and command conflicts, as represented by user workspaces, requires negotiation mechanisms among multiple users, their access devices, the environment, and ongoing computations and services. We believe information management and retrieval services will be improved by utilizing user workspaces. Finally, to build truly responsive rooms, we need necessary techniques to understand the user's commands. A significant effort in combining multiple command sources, such as speech input, gesture recognition, tactile input, and other sensor input is necessary to clearly understand the command and respond in an appropriate manner.

1.3.1 ACE System Architecture

Ambient Computational Environments require significant re-thinking of how applications are developed and structured and the nature of communications among multiple components within the ACE. Today's computing/networking milieu is characterized by client-server relationships between distinct computers and by local applications that interact with their environment through four primary portals: local graphic display, local keyboard and mouse inputs, local file systems, and network connections. One might add a fifth application portal: local temporary storage. To pick-up an application from one computer and deposit it on another and expect the application to continue operating is well beyond today's operating system and programming language constructs. Even to consider re-routing an application's graphic display output and keyboard/mouse inputs to another access point is extremely difficult.

We proposed a threefold approach to building ACE computational components:

(1) Wrapping existing applications to intercept and re-route user input/output streams (keyboard, mouse, and graphic display);

(2) A run-time environment suitable for moving access points for input/output, file system, and network connections; and

(3) Programming language constructs to make the creation of ACE components easier.

As a first step, we considered wrapping existing applications, written for either Linux or Windows NT, to intercept and re-route user input/output streams (keyboard, mouse, and graphic display). This is similar to commercial products such as Netopia's Timbuktu and experimental tools such as Cambridge Research Laboratories' VNC frame buffer replication. The advantage of this approach is that we can re-route access from multiple locations to fixed (existing) applications. This approach provided an early concept demonstration and allowed us to incorporate limited existing applications into the ACE environment.

We also designed and prototyped a runtime environment for controlling ACE devices. This work is described further in [4].

1.3.2 Impact of ACEs on Network Performance, Behavior, and Infrastructure

ACEs use the network in an entirely different manner than traditional client/server/Web systems. The replication of contexts to insure persistent state requires extremely low latency, and extremely fast transactions. The ability to move computational contexts through the network and the need to re-assign input/output streams means that we must develop protocols and routing mechanisms that attend to individual computational sources and computational sinks, rather than physical hosts and application ports. Further, we anticipate ACEs to eventually cross traditional Internet address spaces, a common criteria to allow/disallow communications. ACEs redefine how computational structures operate across networks.

We initially identified three kinds of ACE network transactions that differ significantly from today's network traffic:

(1) ACE traffic will be transactional in nature. There will be short messages that must be delivered in short order, in a reliable fashion, and crossing the wide area network.

(2) ACEs will support multiple, high definition video, audio, and graphic streams; each stream at least two orders of magnitude greater than today's limited streaming video/audio.

(3) ACEs will support highly interactive user input/output systems across the wide-area network.

We anticipate that we will need to carefully consider the network protocols necessary to carry out quick transactions, possibly across a wide area network, to maintain consistency and effect reasonable interactions with the user's contexts. As examples of transactional network traffic, we consider mouse and/or other pointer input, keyboard input, and/or voice command input.

A second, important component of the proposed work was the careful measurement and analysis of the impact of Ambient Computational Environments on the network performance and behavior. The deployment of ACEs by Internet providers will have significant impact on the services which customers expect from the network, and consequently will impact the management and control of the network. For example, customers working from home require access to their work environment. Communication between customers at home and the ACE services at work requires reliability and quality of service from the network far beyond what the typical Internet user receives today. Directory services implemented using protocols such as LDAP with customer information would likely be required to support the allocation of the appropriate network resources to the customer connections.

In this vein, we obtained support from and worked with Sprint to extend the prototype ACE to multiple sites. We designed and build a demonstration prototype (with Sprint support) at Sprint and interconnected the two ACE sites with the Sprint / KU testbed network.

Detailed characterization of the impact of ACE concepts on local and wide area networks was not carried out due to the de-scoping of the DARPA contract.

1.3.3 Tools and techniques to access the ACE

Access to an Ambient Computational Environment is via a Personal Interaction Device (PID). As hypothesized, it comes in a variety of forms to support a variety of interactions styles, abilities, and needs. One version consists of a modest sized color LCD display which supports pen-based input and wireless communication. Advanced devices may as well support voice input. PIDs automatically connect via wireless networks to their local environment, including seamlessly disconnecting and connecting as necessary when the PID is moved from one environment to another. Through this device its owner can interact with all manner of computing environments, including home, office, car, conference room, airplane, bank, grocery store, etc. The PID is responsible for connecting via whatever communication media is available and maintaining its owner's context. In addition, the PID must be able to work with whatever surrounding environment is available, including other displays, display walls, printers, etc.

1.3.4 Programming Languages and Run-time Systems

The implementation of an ambient-based programming language also presents challenges. Programming languages abstract computational execution environments to simplify composition of applications. In a single-workstation or mainframe computer environment, languages and operating systems provide abstractions for file access, I/O devices, memory, CPU cycles, and other resources. The use of local area networks introduced language constructs that provide abstractions for building distributed computations. Abstractions include remote procedure calls, distributed object-oriented programming, client-server architectures, etc. A common goal is a set of abstractions over the network so that all machines can be considered just one hop apart and connected by a reliable communication link.

Current work on programming mobile computers attempts to adapt local area network programming abstractions to richly connected networks. While local area network abstractions are useful, they cannot be directly adapted to these new computational environments. The problem is that existing abstractions suppress the very details that are crucial for implementing robust computations on richly connected networks and mobile computers. Issues that must be considered include:

Location -- What are the computing and communication resources available in the current physical location? What are the resources available in the connected environment?

Access -- How do I identify myself? what authorization do I have to modify the surrounding computing and communication environment?

Reliability -- How does one create a robust, time invariant computation accessible from multiple locations?

For example, a computation must be aware of where it is located and know how to move between locations and communicate with computations at other locations. This is required since richly connected networks are heterogeneous, and resources required by a computation may not exist where it is currently located. As another example, a computation must be able to perceive and react to bandwidth fluctuations in order to accomplish its goals on time; it may be more practical to move the computation elsewhere than tolerate a low-latency communication link. Finally, a computation must be able to gracefully handle failed connections and node failure.

1.4 Task Negotiation among Multiple User Workspaces

We now turn our attention to the user Workspaces. Workspaces relate user intent, action history, and commands with (multiple) computational contexts and the workspaces of other users, all within the ACE. The Workspace describes the user's intent and immediate commands, and constrains action by parameters, such as accuracy of result, timeliness of answer, quality of presentation, quality of source data used, and so on. The Workspace also represents the user to other users' workspaces and aids in establishing working relationships between multiple users and the ACE.

The Workspace/PID combination should recognize: (1) the complete history of its interactions with its owner; (2) the usual desires, wishes, and goals of its owner (an owner model); (3) the current goals of its owner (current task model); and (4) the current PID description (i.e., where am I, what time is it, what are some residual, persistent tasks, when are they scheduled, what else am I doing right now in the background, etc.) Given a request by its owner (the request can be implicit, such as opening a document or sending a fax) the Workspace/PID needs to know why the owner is performing this action.

A user's request for service is formulated with some intent in mind. The request is most easily formulated in terms familiar to the user, e.g. the use of "soda" as opposed to "pop." However, the local environment may only understand a different word, e.g. at KU our dispensing machines only understand the word "Coke." A negotiation must take place to resolve these differences of expression and reach a common level of communication. Likewise, conflicts may occur in an ACE conference room as multiple commands are issued by one or more users. Mechanisms need to be established to resolve these conflicting situations.

Usually the action context will not be defined explicitly. The PID and local services need to recognize the situation and define the action context itself. In cases of uncertainty the PID may ask its owner for clarification, but this should be avoided as much as possible.

1.5 Contextual Information Retrieval

Users roaming through Ambient Computational Environments engage in a variety of tasks: document composition, teaching, meetings, computing, et cetera. As they perform their tasks, they often need to access outside information to support their work. The particular information useful to them is defined by the searches they request and the Workspace, described in the previous section, in which they are working, where the Workspace is defined by their individual histories, the computing facilities currently available to them, and the task they are currently performing. However, existing search techniques are based, almost without exception, on simple word matches between queries and documents.

What is the root of the problem? Almost without exception, current retrieval systems rank documents versus queries using some variant of the ubiquitous (term frequency in the document times inverse document frequency in the database) formula. This has been shown to be extremely effective in practice. However, the empirical evidence has largely come from single, static collections of documents. The dynamic, highly variable Web and the work the user is currently performing requires we extend the model of query versus document matching by explicitly incorporating temporal and workspace knowledge. We need to include the concept of time and context in the queries and the way that document collections are matched to queries.

1.6 Disambiguation of commands within different environments and user contexts

An Intelligent Room, must have some awareness of its state, some understanding of how it might manipulate both its virtual and real states in order to modify them, and some ability to carry out those changes. In addition, an intelligent room must have some purpose or goals that it should strive to achieve or maintain in the face of human and other interactions. Most of these goals would be generated by the occupants of the room thereby necessitating the need for the additional ability of an intelligent room to communicate with its human occupants in order to help them to achieve their goals.

We therefore envision an Intelligent Room as consisting of sensing devices with associated interpretation software, computer software and fast processors for various kinds of reasoning, and devices for effecting the real world environment. In order to have an awareness of its real world environment we envision the sensory devices of an Intelligent Room will include a vision system for seeing what is in the room, hearing systems for hearing what is going on in the room, touch and force sensors of various types placed on various surfaces (such as table tops, floors, and door handles) to feel the weight or pressure of who is in the room. Other sensing abilities such as sonar or laser range finders may also play a role.

Ambiguity resolution is ubiquitous in the human system -- the visual system must resolve illusions created by figure-ground reversals, the proprioceptive system must resolve ambiguities of direction of motion and impact, and the language processing system must complete the comprehension of ambiguous sounds, words, sentences, and references to other system states. In addition, from ambiguous clues the memory system must decide upon a reasonable context in which to reason, and the more conscious reasoning processes must resolve ambiguities due to the default or probabilistic nature of their reasoning. Likewise the actuation system must plan and choose appropriate actions to carry out its goals from among many a myriad of possible ambiguous plans . Moreover, ambiguity resolution in each of these systems is facilitated by allowing interaction with the ambiguity resolution processes of other systems. For example, a mirage might be disambiguated by attempting to touch it, by remembering previous analogous situations, by being informed about the situation, or by inferring consequence from some beliefs about the propagation of light. Likewise a natural language communication might be disambiguated by observing or feeling the external real world in the context of the discussion, by remembering previous assumptions of the speaker in a similar situation, or by inferring conclusions from beliefs.

2 Research Approach

The ACE infrastructure was deployed in six types of environments:

Exploration Environments -- laboratories to develop the underlying ACE technologies as described in the section on Research Descriptions,

Common Conference Rooms -- existing rooms used for project meetings, discussions, and presentations,

A Large Conference Room -- existing room for meetings of up to 100 people, Offices -- a few general offices were equipped,

Laboratory Environments -- existing laboratories used for sponsored research in such areas as wireless systems, adaptive computational systems, light wave network systems, networking systems, design tools, and intelligent information management,

Personal Interaction Devices -- small devices used to identify individuals and interact with the ACE.

We embedded a wide range of devices in these environments. A typical ACE conference room includes: a couple of control computers, video projection system, video cameras, and audio input and output. All devices were controlled from the control computers.

An important question we addressed was: How to securely manage ACE? With embedded computers, cameras, microphones, and other devices in an office or conference room, we wanted to insure only authorized people could control the devices. Authorization implies authentication and security implies confidentially of meetings. Thus, authentication, authorization, and confidentially were key properties necessary for deploying ACE.

To address device management and the security issues, we designed and implemented a device controller daemon. The daemon consisted of a core framework supporting common function across all devices and custom components for each device. Common functions include authentication, authorization, confidential communications, and command processing. Custom functions would involve controlling a camera, or projector, or audio input.

ACEs were built in two development laboratories, four conference rooms, an auditorium, and three offices in Nichols Hall (KU West Campus); in three conference rooms in Eaton Hall (KU Campus); and in a conference room at Sprint Corporation in Overland Park, Kansas. The laboratories are used for software development and system architecture development and the remaining sites are used for deployment and evaluation.

Details of the ACE daemon and common frameworks are available in the technical reports listed in Table 1.

3 Project Structure

During the project we were able to design the major components of an Ambient Computational Environment and implement prototypes for many of the devices. In particular we implemented prototypes to control cameras, projectors, sound input and output, video input and output, fingerprint readers, iButton token readers, a services directory, and workspace access. This work is described in a set of technical reports described in Table 1 and available from the Information and Telecommunications Technology Center.

ITTC-FY2001-23150-01	Renzo Hayashi, Leon Searl Gary Minden	The Ambient Computational Environments Architecture for Reliable, Secure, and Pervasive Computing	December 2000
ITTC-FY2002-23150-02	Renzo Hayashi, Leon Searl Gary Minden	ACE Architecture Design	December 2001
ITTC-FY2001-23150-03	Leon Searl, Gary Minden	ACE General Service Daemon Data Thread, Command Semantics and Client Command Design	January 2001
ITTC-FY2001-23150-04	Renzo Hayashi, Leon Searl Gary Minden	ACE Project Service Command Language Specifications Version 1.0	July 2000
ITTC-FY2001-23150-05	Leon Searl, Gary Minden	Ambient Computing Environment: ACE Service Interface Specification	January 2001
ITTC-FY2001-23150-06	Renzo Hayashi, Leon Searl Gary Minden	ACE Service Directory Interface Specification	May 2001
ITTC-FY2001-23150-07	James Mauro Leon Searl, Gary Minden	ACE Connection Interface Specification Version 0.9	January 2001
ITTC-FY2001-23150-08	James Mauro Leon Searl, Gary Minden	ACE Project ACE Authorization Interface Specification Version 0.1	June 2001
ITTC-FY2003-23150-09	Olaf Landsiedel Gary J. Minden	MPEG-4 for interactive low-delay real-time communication	December 2003
ITTC-FY2004-23150-10	James Mauro Gary J. Minden	Security Model in the Ambient Computational Environment	June 2004

Table 1: the ACE Technical Reports.

3.1 Participants

The project supported the following faculty in EECS and other departments across the university:

G. Minden (PI)	A. Ambler (HCI, Programming languages and systems)
F. Brown (AI systems)	H. Scott Hinton (Advanced Learning Environments)
J. Evans (Networking and Computing Systems)	D. Niehaus (Distributed Systems and Real-time)
C. Tsatsoulis (Expert Systems, agent systems, and case based reasoning)	J. Miller (Graphics)
A. Agah (Robotic Systems)	J. Gauch (Video Information Systems)
S. Gauch (Information Retrieval)	W. P. Alexander (System design)
T. Schreiber (Human Memory)	J. Stiles (Radar)
S. Speer (Psycholinguistics)	C. Allen (Radar and optical systems)
J. James (Distributed Systems)	M. Kong (Algorithms)
J. Sereno (Linguistics)	K. Demarest (Electromagnetics)
A. Jongman (Linguistics)	S. Chakrabarti (Signal processing and neural networks)

Table 2: the faculty members involved in ACE.

The project also supported Mr. Leon Searl and Mr. Dan DePardo, members of our technical staff.

3.2 Research Training:

We placed a strong emphasis on disciplined, engineered system and software design and development. Mr. Searl and Mr. DePardo brought many years of industrial experience to the project and closely worked with the students to develop the student's skills. All students were required to write (and re-write) component requirements, specifications, design, test, and presentation documents. We strongly emphasized the written, communication, documentation, and correct operation of ACE software systems. Students who participated in ACE have a strong grounding in the research development process and necessary skills in team operations.

The following graduate students worked on or were supported by the project: R. Hiroshi, A. Mandapak, B. Rajagopalan, F. Jones, P. Ramasubramanian, R. Kalicut , R. Naraparaju, S. Murugeshan , S. Penumarthy , V. Sankaranarayanan, J. Miadowicz, H. Chen, S. Siddhaye, S. Kalburgie, J. Mauro, and O. Landsiedel.

The following students worked on the project as undergraduates:, J. Mauro, E. Ackers, J. Johnson, C. Chou, J. Bogner, J. Hamilton, R. Andrews, C. Whiteley and B. Yeganeh. The undergraduate students working on ACE participated in The University of Kansas

Engineering Expo in Feb. 2002. They presented a poster session and were recognized as one of the top five presentations.

A number of graduate students supported by the project successfully completed their MS defense. These include:

Renzo Hayashi, MS/CoE, April 26, 2002, *The Ambient Computational Environments Architecture for Reliable, Secure, and Pervasive Computing*.

Sankaranarayanan Vidyaraman, MS/CoE, August 14, 2002, *Security in the Ambient Computational Environment*.

Sivaprasath Murugeshan, MS/CS, December 3, 2002, *A robust Persistent Storage Architecture for ACE*.

Sreenivas Sunil Penumarthy, MS/CoE, December 17, 2002, *Design and Development of a User Level Thread Library for Testing and Reproducing Concurrency Scenarios*.

Rama Krishna Kalicut, MS/CS, January 20, 2003, *Implementation of a Single Threaded User Level Asynchronous I/O Library Using the Reactor Pattern*.

Balaji Rajagopalan, MS/CoE, May 9, 2003, *Statistical Analysis of Human Factor Studies of Graphical User Interface Components*.

Olaf Landsiedel, MS/CS, November 18, 2003, *MPEG-4 for interactive low-delay real-time communication*.

Sachin S. Siddhaye, MS/CS, January 28, 2004, *Rule Induction from Incompletely Specified Data*.

Sandeep Kalburgie, MS/EE, May 4, 2004, *Inducing Strong Rules From Raw Data Sets*.

James Mauro, MS/CoE, May 24, 2004, *Security Model in the Ambient Computational Environment*.

Aditya Mandapaka, MS/CS, December 16, 2004, *KUSAR - A Communication Protocol for Wireless Sensor Networks*.

3.3 Additional Support

The NSF Research Infrastructure grant for ACE supported the following projects:

- DARPA, <u>Architecture and Prototype of an Ambient Computational Environment</u>, ~$1.3M over 30 months (terminated due to budget constraints after 18 months);
- Sprint, <u>Ambient Computational Environments</u>, ~$335K over 12 months;
- NSF, CAREER Award "Cooperative Agents for Conceptual Search and Browsing of World Wide Web Resources," Susan Gauch, $200K, 7/1997 - 6/2001;
- Information and Telecommunication Technologies Center, "The Quality Search Engine," Susan Gauch, $20K, 8/1999 - 8/2000;
- NIMH, S.R. Speer, “Prosodic structure in sentence comprehension", $552K 1995-2000;

- NIH/NLM (Duke University Medical Center), J. Grzymala-Busse, "Informatics Techniques for Medical Knowledge", $137K, 1997-2000; and
- NSF, CISE Research Instrumentation Grant: Ambiguity Resolution for Intelligent Systems Using a Cognitive Robot," Brown, F. M., Agah, A., Gauch, J., Schreiber, T., Speer, S. R., $146K, 1999-2002.

4 Review of Project Accomplishments

The ACE Daemon Framework was designed to provide common services for integrating a large variety of devices into ACE. The framework provided a common command language, authentication and privacy support, authorization policy support, and abstract device support. We have built daemons for two types of pan-tilt-zoom video cameras, projectors, audio input/output, video input/output, fingerprint readers, and iButton token readers. During this past year we have re-written and expanded the daemon framework and improved video and audio capture, transmission, and display.

Resources in ACE consist of devices in a room, the rooms, and the services. The ACE Resource Manager tracks these resources and their status. When the daemon for a device is started, its room location and status are recorded with the Resource Manager. Status is periodically updated. When a service, such as a two-way video link is started, the service can query the Resource Monitor to determine possible devices and their status. We have a prototype of the ACE Resource Manager completed and it is tested.

A robust persistent store has been developed for ACE. The persistent store is used to keep redundant copies of data structures on two or more computers. The computers are linked by Gigabit Ethernet and use active messaging to maintain a consistent state. We have a prototype persistent store operating and it is being used to implement the ACE Service Directory.

User access to ACE resources is through portals or access points. We have experimented with using conventional desktop computers, laptops, and PDAs (Compaq's iPaq). We have used iPaq via wireless IEEE 802.11 networks to carry Voice over IP streams. One can place phone calls from the iPaq while on the move through the ACE environment.

We have a prototype User and Workspace Management system. Users must be registered with ACE to access resources. Once a user is registered, a workspace is created for that user. In the prototype, workspaces are held on Windows-based servers and accessed over VNC. (VNC is a open source frame buffer sharing application available from Cambridge Research Laboratory.) To access their workspace, a user first identifies themselves to ACE at a portal. Current identification techniques are fingerprint scans or iButton tokens. Once the user is authenticated, the location of their workspace is found in the ACE Service Directory and a connection is made to the VNC server hosting their workspace. Their workspace (currently Windows) is shown on the portal. In this case, the portal is usually a desktop or a laptop computer.

"Security" is designed into ACE. By "security" we mean user identification (authentication), authorization mechanisms and policies, and privacy. For authentication we have experimented with fingerprint scanners and iButton tokens. For security policies and authorization we are using the University of Pennsylvania's Keynote systems. And for privacy we are using Secure Socket Layer (SSL) encryption to protect commands and data streams (e.g., the video from an office ACE). These security capabilities are in the prototype stage.

4.1 Year 1 Accomplishments (September 1999 – September 2000)

In the first year of our project we:

- Established a research team,
- Established research development laboratories for ACE technical problems,
- Deployed initial environments for use, and
- Built additional research support for ACE concepts.

4.1.1 Establish a Research Team

We established an experienced and strong research team that is defined the procedures for developing quality systems.

First, we hired Mr. Leon Searl as the system architect. Mr. Searl earned an MSEE from The University of Kansas in 1986, spent several years developing software for TRW Corporation, and has maintained large, commercial software systems. Mr. Searl brought significant software development experience and discipline to our undergraduate and graduate students. Mr. Searl's position was slightly different from our original proposed "Technician" position, but we felt the current position and duties were better suited to the needs of the project.

Second, through ITTC resources, we built an initial research team for development of ACE control elements. We hired a recent KU BSCoE graduate who worked on his MSCoE on ACE and hired five undergraduate students who were first designing and then developing software for camera control, projector control, fingerprint identification, and security mechanisms. These students and other new students were supported under a DARPA NGI grant.

4.1.2 Research Development Laboratories

We acquired two development laboratories: a basic system architecture laboratory and a laboratory for working on disambiguation and speech and gesture input. These laboratories were used for software development and system architecture development. The work in these laboratories was deployed to the larger ACEs we established.

Within our System Architecture development laboratory, we designed and specified a secure architecture for gaining access to and controlling devices, such as video cameras, projectors, sound systems, and fingerprint identification systems.

Within our laboratory for disambiguation, we have deployed systems to support work in artificial intelligence, linguistics, and communications.

4.1.3 Initial ACE Deployment

In year 1 we deployed ACE into the following environments:

- Two conference room settings,
- Two office settings, and
- One large conference room setting.

This is consistent with our proposed plan, however we moved up deployment to the large conference room because of the initial funding level and the visibility the large conference room proved to the ACE project. We anticipate deployment of a sixth ACE environment to a classroom to help support efforts in advanced learning environments.

4.1.4 Build Additional Research Support

The ACE concept represents a new approach to pervasive computing. The NSF grant supported significant infrastructure, but we also needed to develop additional support to develop the software systems and installation of such environments. Additional projects supported by the NSF Research Infrastructure grant were listed in Section 3.3.

4.2 Year 2 Accomplishments (September 2000 – September 2001)

Our goals for our second year were:

- Continue support for research development laboratories for ACE technical problems,
- Establish a research team, and
- Continue deploying ACE environments.

We completed the acquisition of five development laboratories: a basic system architecture laboratory; a laboratory working on disambiguation and speech and gesture input; a laboratory for persistent storage; a laboratory for information retrieval; and a laboratory for audio/video development. One of the conference rooms was deployed at Sprint in Kansas City, approximately 70 KM away.

These laboratories were used for software development and system architecture development. The work in these laboratories will be deployed to the larger ACEs we have started to establish.

Within our System Architecture development laboratory, we designed and specified a secure architecture for gaining access to and controlling devices, such as video cameras, projectors, sound systems, and fingerprint identification systems. This laboratory continues to develop the foundation for the ACE system.

Within our laboratory for disambiguation, we deployed systems to support work in artificial intelligence, linguistics, and communications. Through separate support we worked on integrating robots with the ACE environment.

Within our laboratory for persistent storage we established a small network connected with Gigabit Ethernet and initiated the design and implementation of a persistent storage service.

Within our laboratory for information retrieval we established a small, but large capacity file service system used for studying the time evolution of web pages.

Within our laboratory for audio/video development we have established multi-channel audio and multi-channel video input and output systems that were integrated with our ACE System Architecture.

Mr. Dan DePardo, supported by a DARPA NGI project, supported the installation of ACE systems. He brought professional experience to quality installation and configuration of ACE systems.

Third, we have built a research team to develop ACE . Support for the team was primarily through a DARPA NGI project. In the second year, ACE research projects supported more then 23 graduate research assistants and more than 9 undergraduate research assistants. Several of the undergraduates have expressed their intent to continue their studies in graduate school. Seven of the students were women.

4.3 Year 3 Accomplishments (September 2001 – September 2002)

The goals for our third year were:

- Design, implement, test, and evaluate a framework for integrating a wide variety of devices into an ACE environment.
- Design, implement, test, and evaluate a resource management system for ACE, called the ACE Service Directory.
- Design, implement, test, and evaluate a distributed system for persistent storage.
- Integrate PDAs and laptops into ACE.
- Design, implement, test, and evaluate a framework for user management and workspace management.
- Design, implement, test, and evaluate a security policy management system for controlling access to ACE resources.

These were consistent with our proposed effort.

ACE was deployed in five development laboratories, three conference rooms, an auditorium, three offices, and a remote site. The laboratories were used for software development and system architecture development and the remaining sites were used for evaluation. We started working with PDAs and laptops as access points to the persistent user workspace.

While we were successful in securing additional funding to develop ACE, several contracts are either at the end or were terminated during Year 3. Our work with Sprint Corporation, a one year agreement, ended. We established an ACE at Sprint located approximately 70 km away in Kansas City and successfully linked the Sprint site with those located at KU.

In 2000 we secured a contract with DARPA to develop ACE for the Next Generation Internet Program. This contract supported all the graduate and undergraduate students on the project. However, due to budget cuts, DARPA terminated the contract in January 2002. We have had to let our students go and effective work on developing ACE software stalled. With the loss of the DARPA funds, we re-focused ACE around:

- Providing infrastructure support to faculty pursuing ACE related research;
- Providing support to undergraduate education and research;
- Provide support to a new building that houses the Electrical Engineering and Computer Science Department; and
- Seeking additional resources to support ACE through industry and government resources.

4.4 Year 4 Accomplishments (September 2002 – September 2003)

During Year 4 we focused on supporting EECS and ITTC faculty and research projects. This included acquiring computers and specialized equipment.

We also implemented a second version of the ACE framework and used the new framework to build a network aware video codec. Mr. James Mauro rewrote the ACE framework and presented that work as his M.S.CoE thesis. Mr. Olaf Landseidel (a Fulbright scholar from Germany) used the framework to write an MPEG-4 codec that adapted to network conditions in real-time. He presented his work as his M.S. CoE thesis.

4.5 Year 5 Accomplishments (September 2003 – June 2004)

We requested a no-cost time extension for our ACE project. The project ending date was extended to June 30, 2004.

We supported numerous projects throughout the Electrical Engineering and Computer Science (EECS) Department, The Information and Telecommunications Technology Center (ITTC), and The School of Engineering. The major items are:

- Supported a new file server and back up system for EECS. This provides undergraduate and graduate students with reliable online storage and mechanisms to implement experimental disk partitions for classes such as operating systems and databases.
- Supported a state-of-the-art presentation and teleconference capability for the School of Engineering. This enhances the ability of the school to present material to students, transmit workshops and seminars from the school, and enable students to participate in workshops and seminars from other schools.

- Provided projection equipment, stereo cameras, computers, and projection screen to EECS undergraduates to generate stereo computer displays. The students set up the equipment, programmed content, and explained how the system worked to hundreds of visitors during our School of Engineering Exposition in February 2004. The EECS students won first place for their effort.
- Implemented a teleconferencing capability at ITTC to transmit workshops and seminars and interact with colleagues across the country and world. We have transmitted an NSF workshop on "Community Workshop on Computational Simulation and Visualization Environment for NEES (Network for Earthquake Engineering Simulation)" in December 2003, supported the NSF sponsored "Mobile Sensor Web for Polar Ice Sheet Measurement (PRISM)" project, and interacted with many colleagues across the U.S. and around the world.
- Built an antenna structure to support antennas for several projects including our NSF "Flexible Wireless," NSF "National Radio Networking Research Testbed," and PRISM projects. This enabled us to rapidly and safely mount multiple antennas for research projects.
- Supported "re-tubing" the display projectors for the Design Lab immersive display. DesignLab is a previous NSF Research Infrastructure project. This extended the useful life of this three-projector immersive display system. The display is used by faculty and students in the Mathematics, Geography, and EECS departments.

5 Publications

The following papers, student projects, and book chapters utilized equipment supported by the ACE Research Infrastructure grant. Technical Reports are listed in Table 1. The documents are listed as they were reported in annual reports. Some papers that were "submitted," "under review," or "in preparation" may have been published since those project reports.

5.1 Papers

Alison Alvarez, Stanislaw Bajcar, Frank M. Brown, Jerzy W. Grzymala-Busse, Zdzislaw S. Hippe. "Optimization of the ABCD formula used for melanoma diagnosis," Proceedings of the IIPWM'2003, International Conference on Intelligent Information Processing and WEB Mining Systems, Zakopane, Poland, June 2-5, 2003, 233-240. Springer-Verlag.

Babak Yeganeh completed his Undergraduate Honors research and produced the following report: "Learning from Imbalanced data Sets: A Comparison of Over-sampling and Downsizing Strategies," December 2003,

Brown, Frank, 2004, "Methods for Solving Necessary Equivalences", Proceedings of the Second International Conference "Information Research, Applications and

Education", i.Tech 2004, Varna, Bulgaria - 2004, pages: 28-36, ISBN 954-16-0031-X, FOI-COMMERCE, 2004 Sofia.

Brown, Frank, 2004, "Methods for Solving Necessary Equivalences", Information Theories and Applications, Vol. 11, no. 4, pages: 431-438, 2004. (revised version of the i.Tech 2004 paper with the same name)

Brown, Frank, 2004, "On the Relationships Among Quantified Autoepistemic Logic, its Kernel, and Quantified Reflective Logic", Proceedings of the Second International Conference "Information Research, Applications and Education", i.Tech 2004, Varna, Bulgaria - 2004, pages: 28-36, ISBN 954-16-0031-X, FOI-COMMERCE, 2004 Sofia.

Brown, Frank, 2004, "On the Relationships Among Quantified Autoepistemic Logic, its Kernel, and Quantified Reflective Logic", Information Theories and Applications, Vol. 11, no. 4, pages: 431-438, 2004. (revised version of the i.Tech 2004 paper with the same name)

Brown, Frank, 2004, "Representing "Recursive" Default Logic in Modal Logic", Proceedings of the Second International Conference "Information Research, Applications and Education", i.Tech 2004, Varna, Bulgaria - 2004, pages: 28-36, ISBN 954-16-0031-X, FOI-COMMERCE, 2004 Sofia.

Brown, Frank, 2004, "Representing "Recursive" Default Logic in Modal Logic", Information Theories and Applications, Vol. 11, no. 4, pages: 431-438, 2004.(revised version of the i.Tech 2004 paper with the same name),

Hong Chen completed her graduate research project and produced the following report: "MLEM2 Algorithm for Rule Induction from Data with Missing Attribute Values," December, 2003,

J. R. Miller and S. Yengulalp, Using Jini for Shared Distributed Control of Applications, submitted to Graphics Interface.

J. R. Miller, S. Yengulalp, and P. L. Sterner, A Framework for Enabling Collaborative Control of Applications, submitted to ACM Transactions on Computer and Human Interaction.

Jay Hamilton completed his Undergraduate Honors research and produced the following report: "Interesting Rule Induction Module", December 2003,

Jerzy W. Grzymala-Busse, Linda K. Goodwin, Xiaohui Zhang. "Increasing sensitivity of preterm birth by changing rule strengths," Pattern Recognition Letters, vol. 24, N 6, (2003) 903-910.

Jerzy W. Grzymala-Busse, Stanislaw Bajcar, Witold J. Grzymala-Busse, Zdzislaw S. Hippe, "Data mining analysis of the ABCD formula used for diagnosis of melanoma," Proceedings of the CS&P, 2003, Workshop on Concurrency, Specification and Programming, September 25-27, 2003, Czarna, Poland, 205-212.

Jerzy W. Grzymala-Busse, Zdzislaw S. Hippe, and Mariusz Wrzesien, "Classification of multi-category cases by a binary classifier - methodology and programming tools (in Polish)," Prace Naukowe Akademii Ekonomicznej 275 (2003) 121-127.

Jerzy W. Grzymala-Busse, Zdzislaw S. Hippe, Maksymilian Knap, Teresa Mroczek, "A new algorithm for decision tree generation - a concept and implementation (in Polish)," Proceedings of the 13-th National Conference on Biocybernetics and Biomedicine Engineering, Gdansk, Poland, September 10-13, 2003, 257-262.

Jerzy W. Grzymala-Busse, "A comparison of three strategies to rule induction from data with numerical attributes," Electronic Notes in Theoretical Computer Science 82, N4 (2003), 9 pages, URL: http:/www.elsevier.nl/ locate/entcs/volume82.html .

Jerzy W. Grzymala-Busse, "A comparison of three strategies to rule induction from data with numerical attributes," Proceedings of the International Workshop on Rough Sets in Knowledge Discovery (RSKD 2003), associated with the European Joint Conferences on Theory and Practice of Software 2003, Warsaw, Poland, April 5-13, 2003, 132-140.

Jerzy W. Grzymala-Busse, "MLEM2 - Discretization during rule induction," Proceedings of the IIPWM'2003, International Conference on Intelligent Information Processing and WEB Mining Systems, Zakopane, Poland, June 2-5, 2003, 499-508. Springer-Verlag.

Jerzy W. Grzymala-Busse, "Rough set strategies to data with missing attribute values," Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, November 19-22, 2003, Melbourne, FL, 56-63.

Juan M. Madrid, Susan Gauch. "KeyConcept: Un motor de búsqueda conceptual," Revista Sistemas y Telemática, Universidad Icesi, Cali, Colombia, Jan-Jun 2003. ISSN 1692-5238, pp. 47-62 (in Spanish).

Khashayar R. Baghaei and Arvin Agah. (2003), "Task allocation and communication methodologies for multi-robot systems," Autosoft - Intelligent Automation and Soft Computing Journal, in print.

Kok Meng Pua, Susan E. Gauch, and John M. Gauch, "Repeated Video Sequence Identification and Tracking," Computer Vision and Image Understanding (CVIU), (under review).

Kurt Varmuza, Jerzy W. Grzymala-Busse, Zdzislaw Hippe, T. Mroczek, "Comparison of consistent and inconsistent models in biomedical domain: A rough set approach to melanoma data," AI-METH Symposium, Gliwice, Poland, November 5-7, 2003.

Rachel L. Freeman, Jerzy W. Grzymala-Busse, and Mark Harvey, "Functional behavioral assessment using the LERS data mining system - strategies for understanding complex physiological and behavioral patterns," Journal of Intelligent Information Systems 21 (2003) 173-181.

Robert L. Dollarhide, Arvin Agah, and Gary J. Minden. (2003), "Evolving controllers for autonomous robot search teams," Artificial Life and Robotics Journal, in print.

Ron Andrews and Chris Whiteley completed their Undergraduate Honors Research and produced the following report: "Data Mining Experiments Using C4.5 Rule Induction System with a Melanoma data Set," May 2003,

Sachin Siddhaye completed his graduate research project and produced the following report: "Rule Induction from Incompletely Specified Data."

Sandeep Kalburgie completed his graduate research project.

Stanislaw Bajcar, Jerzy W. Grzymala-Busse, Witold J. Grzymala-Busse, Zdzislaw S. Hippe, "Diagnosis of melanoma based on data mining and ABCD formulas," Proceedings of the HIS 2003 - the Third International Conference on Hybrid Intelligent Systems, Melbourne, Australia, December 14-17, 2003 Design and Application of Hybrid Intelligent Systems, ed., by A. Abraham, M. Koeppen, K. Franke, IO Press, 2003,614-622.

Susan Gauch, Jason Chaffee, and Alexander Pretschner, "Ontology-Based User Profiles for Search and Browsing," User Modeling and User-Adapted Interaction: The Journal of Personalization Research, Special Issue on User Modeling for Web and Hypermedia Information Retrieval, Guest Editors: Peter Brusilovsky and Carlo Tasso, (under review).

Travis Wade, Deborah K. Eakin, Russell Webb, Arvin Agah, Frank M. Brown, Allard Jongman, John M. Gauch, Thomas A. Schreiber, and Joan Sereno. (2002), "Modeling recognition of speech sounds with Minerva2," In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002), Denver, Colorado, September 2002.

Zdzislaw S. Hippe, Stanislaw Bajcar, Piotr Blajdo, Jan P. Grzymala-Busse, Jerzy W. Grzymala-Busse, Maksymilian Knap, Wieslaw Paja, Mariusz Wrzesien, "Diagnosing skin melanoma: Current versus future directions," TASK Quarterly 7 (2003) 289-293.

5.2 Chapters in Books

Jerzy W. Grzymala-Busse and Rachel L. Freeman. Improving rules induced from data describing self-injurious behaviors by changing truncation cutoff and strength. In Rough Set Theory and Granular Computing, ed. by A. Skowron and L. Polkowski, Physica-Verlag, 2003, 177-185.

Jerzy W. Grzymala-Busse and Wojciech Ziarko. Data mining based on rough sets. In Data Mining: Opportunities and Challenges, ed. by John Wang, Idea Group Publ., 2003, 142-173.

Jerzy W. Grzymala-Busse, Witold J. Grzymala-Busse, Linda K. Goodwin., and Xinqun Zheng. An approach to imbalanced data set based on changing rule strength. In

Rough-Neuro Computing: Techniques to Computing with Words, ed. by S. K. Pal, L. Polkowski, and A. Skowron, Springer-Verlag, 2004, 543-553.

Jerzy W. Grzymala-Busse, Zdzislaw S. Hippe, Maksymilian Knap, and Teresa Mroczek. New programming tools for knowledge engineering and machine learning. A comparison of some representation systems of uncertain knowledge (in Polish). In Knowledge Engineering and Expert systems, ed. by Z. Bubnicki and A. Grzech, Technical University of Wroclaw Press, 2003, 239-247.

Jerzy W. Grzymala-Busse, Zdzislaw S. Hippe, Piotr Blajdo and Mariusz Wrzesien. New programming tools for knowledge engineering and machine learning. Optimization of some parameters of data mining systems (in Polish). In Knowledge Engineering and Expert systems, ed. by Z. Bubnicki and A. Grzech, Technical University of Wroclaw Press, 2003, 15-21.

Jerzy W. Grzymala-Busse. Data with missing attribute values: Generalization of indiscernibility relation and rule induction. Submitted as a chapter to the book Rough Set Theory and Applications ed. by J. W. Grzymala-Busse, B. Kostek, R. Swiniarski and M. Szczuka, Springer-Verlag.

Jerzy W. Grzymala-Busse. Three strategies to rule induction from data with numerical attributes. Accepted as a chapter to the book Rough and Fuzzy Reasoning ed. by D. Dubois, J. Grzymala-Busse, M. Inuiguchi and L. Polkowski, Physica-Verlag.

6 Conclusion

The NSF Research Infrastructure for Ambient Computational Environments enabled us to explore a new concept in computation: that rich computational resources will be embedded in our environment. We were able to establish a number of ACE rooms in laboratories, offices, conference rooms, and classrooms. The project supported 23 faculty, 16 graduate students, and eight undergraduates.

The ACE RI grant supported a leading edge infrastructure at the Electrical Engineering and Computer Science Department and The Information and Telecommunications Technology Center that continues to support our education and research mission.

7 References

[1] Norman, D. A., The Invisible Computer, The MIT Press, Cambridge, MA, 1998.

[2] Kozyrakis. C. E, and Patterson, D. A., "A New Direction for Computer Architecture Research," IEEE Computer, Vol. 31, No. 11, November, 1998, pg. 24-32

[3] Waldo, J. "Jini™ Architecture Overview," Sun Microsystems, Palo Alto, CA, 1998.

[4] Hayashi, R., Searl, L, and Minden, G., “The Ambient Computational Environments Architecture for Reliable Secure, and Pervasive Computing,” December 2000, ITTC Technical Report ITTC_FY2001-TR-23150-01.