

# **A Framework for Automation Using Networked Information Appliances MIT9904-05**

## **Proposal for 1999-2000 Funding**

**Srinivas Devadas and Larry Rudolph**

### **Project Overview**

The proliferation of diverse and heterogeneous information appliances has produced a new set of problems, challenges, and opportunities. We propose to build a development infrastructure for **information automation** that combines the synergistic use of sensors, standard hardware interfaces, intelligent software, and adaptive algorithms in order to rapidly produce intelligent, networked information appliances. Using our infrastructure, we propose to develop many smart appliances, such as an intelligent navigation system that tells a driver the fastest way of getting from point A to point B, taking into account current traffic and weather conditions as well as driver preferences; a smart thermostat that regulates temperature within a home taking into account human presence and activity; and a smart music system that will search and download music from the web taking into account the user's tastes and preferences and which will automatically play appropriate music based on time and level of human activity.

We envision a computing platform that consists of locally networked computational elements attached to sensors and actuators. The sensors provide image, audio, pressure or position input. The computers control actuators within appliances and also interface with other appliances. Not only is intelligent application software fully customizable and adaptable to user preferences, but it also makes use of our middleware that seamlessly integrates new custom or commodity sensors and actuators. This framework of computers, sensors and intelligent software allows for the rapid deployment of a wide variety of information automation problems. We believe that the *deployment of such a solution should take no more than a week*, including customizing hardware and writing software applications.

### **Composable Computing**

Our approach is based on the (idealistic) notion of composable computing. Providing a single interface for all the available black boxes enables composable computing. Computational elements, sensors, actuators, or networking devices all plug into the same interface so that a user can make use of the devices that

suit the application. The user is unaware of the processor, operating system, or the type of network, and is solely focussed on managing information flow, information transfer, or information processing. As an example, the user may put together a computer block, a network block, and a camera block to create a networked digital camera. Adding image recognition software that is downloaded via the network and executed on the computer can quickly create an intelligent surveillance system. An important feature of a composable computing platform is modularity which allows the user to upgrade any component of the system, be it hardware, or software, or appliance-specific components, independent of other components.

We are implementing a composable computing platform for the rapid deployment of networked and intelligent information appliances. Our composable computing platform consists of a hardware platform, middleware, and a software platform. We describe each of these in the sequel.

### **Hardware Platform and Infrastructure**

Unlike many ad-hoc approaches, our hardware platform consists of credit-card-sized computer cards, and PCMCIA peripheral building blocks that each have *exactly the same interface* which corresponds to the Cardbus™ protocol. The peripherals can be audio cards, interface cards such as IEEE 1394, camera cards, Ethernet or wireless network cards, or even a disk. This symmetry enables composability and modularity. In general, any appliance or set of appliances can be made intelligent or networked using our approach. We envision interconnecting these appliances using wireless, Ethernet or local-area networks, and also providing local computation for each appliance to reduce bandwidth requirements.

A two-socket backplane that can be embedded into any kind of appliance forms the basis of hardware "glue". Appliance shells that provide an easy-to-use or familiar user interface can thus be made more intelligent by attaching an embeddable backplane to the shell. The appliance-specific component can be upgraded independently of the specific cards that are plugged into the backplane socket, as can the software running on any of the cards.

We have built a prototype computer card that incorporates a StrongARM™ processor and 32MB of DRAM as a back-end. The front-end interface appears as a DRAM chip to the back-end processor and memory system. The front-end is currently a Xilinx 4062 FPGA and implements an interface controller, which converts data from the memory bus to the Cardbus protocol. The prototype backplanes we have built correspond to an 8-socket stand-alone backplane, and a 2-socket embeddable backplane. The stand-alone backplane can be used to network a large number of peripherals, disks, or memory with multiple computer

cards. These backplanes provide high-speed interconnect of up to 6 Gbps between the sockets, and between the appliance shell and the sockets.

### **Middleware**

In order to achieve intelligent appliances, intelligent software is required for the locally attached computers. In addition to the usual multitasking functions, the middleware provides a backplane interface so as to communicate with the processor on a computer card and to compensate for the lack of a Cardbus bridge on our computer card. This backplane interface packages requests from devices or device drivers into packets that are understandable by the interface controller FPGA. The backplane interface allows us to use existing 3<sup>rd</sup> party device drivers *without modification*, which is critical to being able to easily use the entire gamut of available PCMCIA peripherals.

### **Software Platform and Infrastructure**

We are implementing a software development environment for rapid development of intelligent applications, especially for the code that is to be executed by the local computer cards. We wish to layer evolving control over distributed applications that implements a watch-reason-automate loop. In the Watch step, data is monitored using different kinds of hardware and software probes. In the Reason step data is analyzed and correlated with past data. The Automate step actually makes control decisions.

Working with the development environment, the programmer implements this watch-reason-automate loop, and this code can be quickly developed through the use of cross-domain intelligence mechanisms (CDIM's). A CDIM can be a pattern recognition engine, a correlation engine or a data-mining engine. These CDIM's enable code reusability at the algorithmic level. While the data streams being processed in different appliances such as an intelligent surveillance system or a route-prediction system can be vastly different, intelligent software in these appliances can reuse the same algorithm-level software. Finally, the user is allowed to guide the automation process and to override decisions, and this is accomplished by providing appropriate hooks in the evolving control module.

This framework will be used to rapidly deploy customized, efficient solutions to information automation problems. Moreover, we believe this framework will better manage the multitude of information streams that are constantly flowing in our environment and will improve human productivity in the long run.

### **Status, Schedule and Demonstrations**

**July 1, 1999:** Prototype hardware corresponding to computer card, 8-socket stand-alone backplane and 2-socket embeddable backplane has already been

built, and will be debugged by this date. The prototype hardware uses FPGA's and therefore its size, especially that of the computer card needs to be reduced.

**December 31 1999:** Linux port to computer card will be complete, and the middleware will be completely written and debugged. Second version of computer card with form factor similar to a PCMCIA card will be complete. We envision fabricating 20 or more computer cards so a large number of demonstrations of intelligent and networked appliances can be performed. A demonstration of a smart remote control that controls multiple appliances including a TV and a stereo and which learns users' preferences through repeated use will be made. The software running on the remote will be written utilizing CDIM's, and will be easily and rapidly modifiable.

**June 2000:** The interface controller chip on the computer card will be fabricated as an ASIC rather than an FPGA to achieve the exact PCMCIA dimensions. This ASIC will enable networking capabilities so demonstrations of networked, intelligent appliances, such as an intelligent GPS system, and a smart, networked audio player can be made.

**June 2001:** In the second year, we will concentrate on intelligent software for automation as well as the integration of more complex sensors into the framework. We will explore the integration of sensors such as NTT's eye mark recorder, the use of networked speech servers from Victor Zue's group, and image recognition software from Eric Grimson's group.

### **Collaboration with NTT Researchers**

We envision collaborating with NTT researchers in three different ways. Firstly, NTT hardware can be used as building blocks to create intelligent systems. For example, the networked audio card developed at NTT Yokosuka Labs can be used to create a rich set of information appliances, including an intelligent PDA that learns the user's music preferences and automatically downloads music that the user enjoys. The integration and use of sophisticated sensors such as the eye mark recorder will be investigated in the second year. Secondly, intelligent software applications can be deployed in collaboration with NTT researchers using CDIM's that we will develop at MIT. For example, the intelligent navigation system will require methods to analyze and predict traffic patterns in real time, and the intelligent player will require methods to analyze and predict music taste. Finally, once our hardware and software infrastructure is built, we will make it available to NTT researchers who may be interested in using the hardware for the purposes of information automation.