Submitted to CACM, REV 2-1. Feb. 2000

Embedded Computation Meets the World-Wide-Web

Gaetano Borriello
Department of CS & E, Box 352350
University of Washington
Seattle, WA 98195-2350 [USA]
gaetano@cs.washington.edu

Roy Want
Xerox PARC
3333 Coyote Hill Road
Palo Alto, California 94304 [USA]
want@parc.xerox.com

ABSTRACT

Two important trends are converging to bring about a radical transformation in the operation of our world. First, the computer industry's remarkable ability to squeeze more and more transistors into a smaller and smaller area of silicon is increasing the computational abilities of our devices, while simultaneously decreasing their cost and power consumption. Second, the proliferation of wired and wireless networking spurred by the development of the world-wide web and demands for mobile access are enabling low-cost connectivity among computing devices. It is now possible to connect not only our desktop machines, but every computing device into a true world-wide web that links the physical world of sensors and actuators to the virtual world of information utilities and services. What amazing new applications and services will result? How will ubiquitous computation affect our everyday lives? Will the long envisioned invisible computing paradigm finally be possible? This paper explores these questions while providing an overview of the capabilities of the new wave of embedded devices.

INTRODUCTION

We live in an opportune time. Never before have so many supporting technologies been available to construct a network infrastructure that pervades everyday life on such a massive scale. Furthermore, over the last 10 years, we have migrated many of our work practices to electronic media. Even mundane consumer products such as ovens, toasters and dishwashers are automated through embedded computation. In fact, 98% of the computing devices sold today are embedded in products whose use does not confront the user with their presence. A new revolution is about to happen, one in which we will gain immense additional value, by connecting all these computational components together [11]. This opportunity confronts us with some important challenges for building useful services, designing more robust and easily manageable systems, and guaranteeing user privacy and security.

The Internet has been the most vital component that has facilitated this opportunity. However, it is worth remembering that any new opportunity requires many components to be in place before a revolution can really happen. For instance, the collection of networks that formed the early Internet, such as NFSNET and ARPANET, were created as long ago as the 1970s. The name Internet was not coined until the mid-1980s as it began to be thought of as one network, unified by the TCP/IP protocol set. It became the network of choice for the rest of the world by 1994 because of the simplicity that the World Wide-Web (WWW) protocols brought to play. Only then was it rapidly adopted outside of its traditional havens, the university computer science department and corporate research laboratories. This expanded user community in turn fueled the expansion of the Internet itself. The larger the Internet grew, the larger the customer base and the more attractive it became to individuals and businesses alike. The trend continues with an important additional factor accelerating the process: Moore's law – which predicts the number of devices that can be fabricated on a chip, doubles every 18 months [5]. The implementations of the TCP/IP protocol of the mid-1980s required the computational resources typically found on a desk or workstation class machine. The embeddable microprocessors of that time were considerably less capable and the ability to support full protocol stacks at a reasonable performance was beyond the scope of these under-powered computational engines. Internet connectivity was thus limited to costly and measured resources.

Fifteen years later, in the year 2000, after 10 cycles of Moore's law, a micro-controller that costs just a few dollars, in combination with one megabyte of memory, is just as capable as a desktop computer in 1985. Such a device can support a compact embedded operating system, can interface to a 10Mbps network, can run a TCP/IP stack and a web-server interacting with the now ubiquitous HTTP protocol. We can expect to see further advances in the miniaturization, and the reduced power consumption of these components. But even now the power budget for these devices can be as low as 50mW. This is sufficiently small that portable devices can be powered from batteries. Further utility can be achieved by replacing wired networking with wireless. Wireless connectivity, which is also possible at a reasonable power budget, enables ubiquitous connectivity and a world in which everything can be connected to a global network. New standards and mass-produced transceivers are driving the cost of wireless connectivity down to levels comparable with the cost of microcontrollers.

These trends point to a new world in which we exploit the synergy afforded by literally billions of interconnected devices, thousands per person. These will be embedded universally across our working environments, and their modular composition will more efficiently facilitate many tasks that require relatively expensive monolithic solutions by today's standards.

COMMUNICATION TECHNOLOGIES

First, we will turn to the technologies that will enable this revolutionary reorganization of our information systems. Standardized ubiquitous protocols are the liberating component for most of these systems. They gather information, deliver it to user services over wired and wireless networks, and present distilled information and events to the user.

EMBEDDED WEB SERVERS

What benefit do we derive by embedding a web-server into an appliance? The basic functionality of the web enables client programs, or browsers, to fetch web pages (files in HTML format) and display them in a browser window. Hyperlinks within a file can further reference other files, either local or remote to that site. More importantly, in the case of an appliance, a link may also reference a Common Gateway Interface (CGI) script. This script executes and returns HTML to the browser. Because the script can generate HTML dynamically, it may incorporate real-time data that is being derived from sensors. Thus any appliance connected in this way can be monitored through a CGI script and the results presented to a user in a convenient graphical form. Similar mechanisms can also be used to directly control an appliance from a remote browser.

Recently, there has been something of a competition between industry and hobbyists alike, as to who can build the smallest web-server. For many of our readers it will not be surprising that a web server can be made small enough to fit in the palm of your hand. Figure 1 shows a web-server designed at Xerox PARC, in 1998, with the purpose of exploring the applications space for embedded computation in the office environment. It has connectors to attach to a 10baseT Ethernet, a serial line, and/or the general-purpose I/O pins of its micro-controller. The device runs the Spyglass web-server on top of the VxWorks operating system. 16MB of DRAM and 1MB of flash memory are available. An alternate software architecture developed at the University of Washington uses uCLinux and a public-domain HTTP server. We mention these details to emphasize what is possible in such a small form-factor which is, to a large part, dominated by connectors. Computational power is no longer the constraining factor. It is also worth emphasizing that this embedded web server, with the stated memory capacity, is capable of serving up volumes of web pages, images and related documents, thus operating as a self contained web-site – in the palm of your hand.

Figure 1. Hydra: Xerox PARC's embeddable web server.

Commercial examples of embeddable web-servers are beginning to appear, such as Dallas Semiconductor's Tini, (see Figure 2), which further illustrates the small computational footprint compared to the adapter board and its connectors.

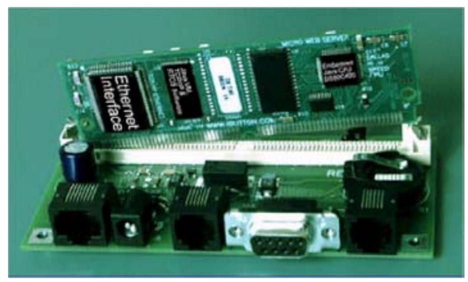

Figure 2. Dallas Semiconductor's Tini web server.

However, some web-server designs are aiming at a totally different space. At the other end of the web-server spectrum is the Boolean server (see Figure 3) whose sole purpose is to turn on a bit, for example, a light-bulb on or off, or to sense the state of a switch, perhaps as part of a security system. At one side of a web-server that implements 'bit' control, or sensing, the system is very simple. But on the network side the web-server must use the same protocols as any other more advanced server.

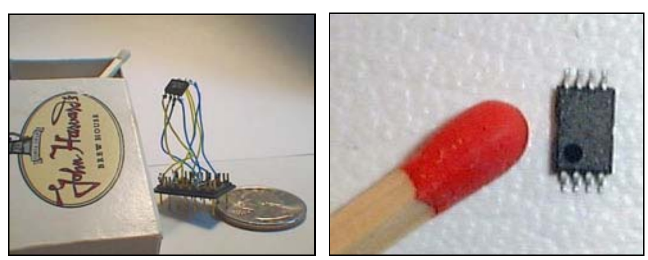

Figure 3. A web-server on a PIC processor (left), and even smaller, a Fairchild ACE1101MT8 (right).

The challenge of the micro-web-server is to implement as little as possible of the HTTP/TCP/IP protocol stack in order to meet the protocol standards, but remain small. These implementations are often called 'slim-servers'. Some of the complexity is reduced by having pre-computed packets that are transmitted by a simple state machine in response to received packets. In this way the computation is off-loaded to the time the server is created, by exhaustively calculating all the different responses that it needs in order to communicate, or acquire, its state. This simplification only makes sense when the state machine is very small. Servers whose complexity exists somewhere between a fully functional web server and a slim-server must use a more skillful implementation of the full protocol suite. As Moore's Law continues to increase memory capacity and computational capabilities while decreasing power consumption, this approach will make sense for an ever increasing range of systems.

JAVA, APPLETS, AND JINI

The CGI mechanism described earlier has its limitations in that all interactions between the user and the web page must return to the server to be processed. Java provides a way of bringing computation to the client so that many interactions can have faster response times. A link embedded in a web page points to a Java 'applet', which is loaded into the browser. The program is composed of Java byte codes intended for execution by a Java Virtual Machine, or JVM. Once received, the code can execute locally in the secure environment of the local JVM. From the standpoint of embedded processing in dedicated appliances, Java applets enable a device to export its interface to a secondary machine that is either near-by, or in another part of the world. Furthermore, depending on the nature of the client, the interface can be customized to suit a specific need, namely, to provide user interaction or control by another program or automatic agent. The Java model is key to opening up convenient interaction between all forms of embedded computation.

Being able to interact with an appliance is only half the battle, the other half is knowing the types of appliances and services that are at hand. Jini [8] is an example of a 'discovery' service and was invented to enable local appliances, or services, to be located by client processes in order to form ad hoc communities of devices that can communicate and benefit from mutual interaction.

A Jini 'lookup' service runs on a local server and acts as a clearinghouse where services and devices register or 'join' and others come to find out what is available. Multi-cast network protocols are used to efficiently locate a lookup service. When a device registers it can also provide Java code that a future client can use to communicate with it. A client that discovers the service also loads this code, a sort of device driver that implements all aspects of device specific code and protocols needed to use the device. The client can then communicate directly with the device without having to go through the Jini service any longer. Java and its mechanism for Remote Method Invocation (RMI) are key technologies that enable uniform access across all services. The Jini service is ultimately only an enabling component to achieve the desired interaction between client and service.

Today devices that rely on small micro-controllers, or those particularly limited in their power budget, are unable to support a JVM. Proxy mechanisms can be used to delegate Jini protocol interactions to surrogate processors that are more capable. Again, Moore's Law ensures that the number of situations for which this will be necessary will be forever decreasing. The research community will be facing major challenges in developing truly universal methods for plug-and-play that will permit all these highly capable devices to aggregate their capabilities and provide interesting services to the user as automatically as possible.

WIRELESS CONNECTIVITY

Until now we have assumed that clients of embedded computation can contact a target service without significant difficulty. In a wired networked world this is a fair assumption, but if we are to fully realize the benefits of ubiquitous embedded computation, many components will not have a physical connection.

Wireless connectivity between embedded devices is extremely desirable, allowing unencumbered mobility and dynamic ad hoc connections between devices. Bluetooth [2] is a recent initiative to provide a low-cost wireless solution for connecting components that are in proximity, separated by several meters. Low-cost will be achieved by complete integration of all the required analog and digital components onto a single mixed-mode chip. By simply adding an antenna, and a minimal set of discrete components, the transceiver will be ready to interface to a digital bus. The most recent implementations are built from a two chip-set. The Bluetooth system has been designed to operate as a spread spectrum device in the 2.4GHz unregulated ISM band. It will use the frequency hopping technique switching between 79 frequencies separated by 1MHz, and the maximum hopping rate will be about 1600hops/s. The system will provide a raw data-rate of 1Mbps, translating to an application level data-rate of somewhere around 721kbps. These are respectable speeds for many applications (compare this with today's ubiquitous web surfing tool, the 56k modem) and for this reason it is expected to be an attractive proposition for application builders.

The Bluetooth consortium has established both the hardware and protocol standards for this new technology. The system is more than a paper design with live demonstrations of Bluetooth being given at conference events, such as COMDEX'99. Nokia and Motorola have shown their cell-phone and laptop products using Bluetooth to synchronize contact lists and transfer files. There are still many opportunities for design growth in this area and the coming years will be crucial to the spread of this nascent standard.

Infrared communication, as standardized by the Infrared Data Association (IrDA) [4], is also a potential candidate for linking embedded computers. At data rates from 9600bps to 4Mbps, there seem to be as many opportunities for applications of IrDA as there are for Bluetooth. However the standard has been in existence for several years and although hardware support is often ubiquitously available on mobile computers, the application level software has had problems associated with interoperability due to the many operating modes that the standard tries to encompass. Infrared also requires, in most cases, line of sight operation. This can be seen as an advantage for those applications requiring explicit selection of a device for which communications is desired (similar to pointing a universal remote control at a particular piece of audio/visual equipment). However, there are other examples for which proximate non-line of sight is also desireable.

Other wireless communication approaches are also likely to find a place. For example, body-based communication schemes that send extremely low-power data signals through a user's skin can be advantageous for private communication and selection of devices by touching or holding rather than just proximity as afforded by RF-based systems.

Thus, many types of wireless communication will likely co-exist. This will necessitate the development of protocols that will support data movement across these heterogeneous networks. In addition, protocols and operating systems will have to be developed that can support intermittent connectivity. Power-limitations (which are only partially mitigated by Moore's Law) and short-range communication (necessary for maximize bandwidth per unit volume and, possibly, for privacy considerations) will mean that devices will not be continuously connected to the wired infrastructure. Proxies, caches, and active networking are some of the technologies that will have an important place in this world of interconnected devices.

DEVICE TECHNOLOGIES

Computation in isolation from the world has limited impact. Myriad input/output devices have been or are being developed to connect users to the computational infrastructure. These range from high-resolution wall-mounted displays to palm-top stylus-based PDAs to conversational speech-based interfaces. However, in order to benefit from the full value of embedded computation it must be possible to sense and control the world directly. Until recently, deployment of sensors and actuators throughout the environment has been prohibitively expensive for two principal reasons: the expense of the devices themselves and the expense of their interconnection, which traditionally has meant a wired connection to a fully featured network. Today, devices are getting cheaper due to new technologies and economies of scale. Connectivity will be wireless and, in many case, intermittent (as discussed in the previous section). Thus, many barriers have been lifted and we can now contemplate an extensive interface between the physical world and our virtual world of information and computation.

MEMS SENSORS

MEMS, or micro-electromechanical systems, are an important solution to sensing that integrates with computation and communication at low-cost and high accuracy. MEMS' sensors are made from novel mechanical structures constructed directly from silicon. They can be made to the same tolerances that are used in the semiconductor industry and because silicon is very strong and robust at the micrometer scale, MEMS sensors are resilient and have considerable longevity. In fact, if single crystal silicon components are used, they can be flexed back and forth for many millions of operations without material fatigue. A common commercial application of MEMS is the accelerometer [1] (see Figure 4) that precisely controls the deployment of the airbag in a modern car. MEMS' structures will eventually become the technique of choice for designing embedded systems that require computation, sensing and control in inexpensive and reliable high-volume production. Their capabilities are being extended to chemical and magnetic field sensing in addition to forces and light levels. It is conceivable that we will soon have microscopic laboratories that can analyze the liquids and atmospheres in which they are immersed. This will spur an age of personalized environmental sensing in addition to communication and computing.

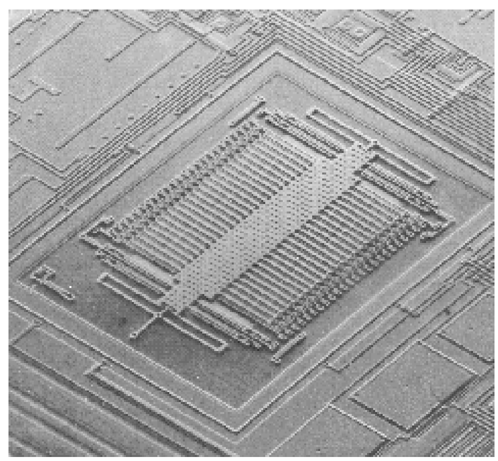

Figure 4: Photomicrograph of Analog Devices MEMS accelerometer.

TAGS

The automatic identification industry, which is not directly associated with embedded computation, has been pushing the limits of computation and miniaturization for the purpose of designing electronic tags for tracking everything from courier packages to livestock. RFID is the name given to an electronic tag (e-tag) that is inductively powered by the tag interrogator. The captured energy is used by its miniature electronics to send its identity (a unique number) back to the interrogator, through a modulated carrier.

The e-tag industry is benefiting from the same advances in lithography that is driving the computer industry and modern tags are becoming quite sophisticated. They now contain on-board memory that can be written to, or read from, by the interrogator and some technologies have anti-collision mechanisms to allow multiple e-tags to be read in the same space, for example, TI's Tag-it product [7] (see Figure 5). Moreover, sensors are also being integrated with e-tags that permit real-time sensor data to be read and returned along with the ID.

Figure 5. Texas Instruments' Tag-it system.

It is clear that e-tags will soon support greater computational functionality. At some point, they may be able to offer a full web-server capability. However, in the near term, if e-tag interrogator technologies have an interface to the Internet, a tag can be effectively part of the network infrastructure while proximate to the reader.

LOCATION TRACKING AND SENSING

While the global positioning system (GPS) can be used to provide high-accuracy location data to a user within line-of-sight of several GPS satellites, indoor location sensing is only beginning to receive serious attention. Tagging technologies can be used, not only to detect the presence of an object, but also its position. By seeding the environment with enough interrogators, tags can be tracked as they move through a space. RF tags who's signal-strength can be measured by base stations can be triangulated-on within a known coordinate system.

Important issues are the resolution of the position information and who has access to it. Different technologies will be required for different applications. It is quite different to know the position of a person as they move through an office environment than it is to precisely track a folder as it moves from filing cabinets to desktops to briefcases. Privacy concerns loom large with location tracking. Approaches that provide users with the ability to determine a position and then use the information as they see fit will have an important place in addition to straightforward tracking approaches where the system keeps track of where the user is. Examples of this dichotomy have existed from the beginning in ubiquitous computing with the Olivetti Active Badge [9] juxtaposed against the Xerox PARCtab system [10].

APPLICATIONS

Distributed sensors and actuators connected through the standard protocols of the world-wide-web and wireless communication media provide a powerful toolkit for developing rich applications that will affect all our lives. In this section, we present three scenarios that utilize the various technologies we have described up to this point.

HOME AUTOMATION

A long sought after vision of the future has been the automation of the home. Many attempts have been made to create 'smart houses', but few have been attractive models that make a compelling case for people to want to live in them. We would argue this has been mainly the result of poor return on investment, at a time when only a few products had any embedded computation at all. Now we see more and more applications of embedded processing, as microprocessors are able to perform a wider range of tasks and exploit standards that present some measure of interoperability. A classic example is the digital camera that is beginning to replace traditional photography. The quality of imaging and post-processing is beginning to compete with an industry that has been established for many decades. At the same time we have a new opportunity of using photographs in a way that has not been possible before. For example, displaying them on a television, or an electronic picture frame, or sending them by email to a relative. Recently we have seen the emergence of services that store and catalog our photographs as well as organizing them into graphically rich web-based photo albums.

We now live in a world in which we can expect common devices to inter-operate. Organizations such as HomeRF [3] are proposing standards to further enable interactions beyond those that would develop slowly in a market economy. Similar interactions can be found in the business environment with computers, PDAs, scanners, printers, computerized whiteboards, cellular phones and automated document tracking. Bluetooth, IrDA, and LANs such as the IEEE 802.11 wireless LAN standard, have focused on this market and businesses are likely to reap the benefits of ubiquitous embedded computation before the home. However, at this point in time, the home environment holds many challenges. We can start with the problems caused by the numerous IR-enabled remote control devices we have around the home. There is another one with virtually every appliance we buy. Wouldn't it make more sense to have generic, touch-screen based remote control devices, that look like palm-sized PDAs, which can upload the user interface from the appliance itself? Then we would just have to make sure we had one of these in every room. Of course, this requires a more horizontal model in which new standards permit appliance-independent descriptions of user interfaces, rather than the current vertical model that forces us to use the specific remote control device that came with the appliance. The universal connectivity afforded by the Internet, along with embedded web-servers, will enable a new way of interacting with our living spaces. Environmental parameters and audio/visual displays will adjust to the occupants of a room and provide levels of energy efficiency not possible before.

A project at the University of Washington [6] seeks to develop a virtual neighbor for elderly relatives. Sensors typically found in the home would be used to collect data on traffic patterns and resource usage so that a remote observer, or automated agent, could ensure that nothing had gone awry. Anomalous situations such as doors, or windows, left open overnight, or the lack of motion around the home for an extended period of time, would trigger messages to those concerned. Crucial to this wide range of new applications is the ability to easily integrate new devices into the home information systems. It should be possible to dynamically add or remove sensors from the home without having to reconfigure or maintain software as we do today (upgrading of new software, debugging malfunctions, etc.). This leads to a plug-and-play model where sensors are as small and cheap as possible, so that they can be deployed in large numbers, and the storage for the data they collect along with the programs that look for patterns on that data, reside at nodes in the wired infrastructure. An embedded home web-server acts as the connector between the wireless sensors and the infrastructure. Code for displaying sensor data, and fusing data from multiple sensors, can be developed by third parties and linked with the data through self-describing mechanisms such as XML [12]. Thus, a remote user monitoring a home through a web browser would see the graphical rendering of the sensor data being gathered. Furthermore, the manufacturer of the sensors, or the third-party developers, would ensure the longevity of these systems by continually updating the software components responsible for data fusion.

The single most important challenge in this application domain will be making sensors, actuators, and the services that use them truly trivial to deploy. Doing this while maintaining privacy will not be an easy task. New approaches to security and access domains will be needed.

EXPERIMENT CAPTURE

Sensor infrastructure also finds many applications outside the home. One that will prove to change the way scientific results are collected and disseminated is embodied in the Labscape project at the Cell Systems Initiative at the University of Washington. Scientists seeking to understand the inner working of the cell are finding themselves hampered by the limitations of the current methods for disseminating research results. There are three main obstacles. First, there is no unified model that integrates our collective knowledge of cell chemistry and mechanics. Second, the experiments that are conducted are not completely captured or are recorded ambiguously thus making them hard to reconstruct. Finally, the overwhelming majority of experiments are never published and serve only to enrich the experience of a handful of researchers rather than the larger community.

The Labscape project seeks to instrument a cell biology laboratory to the point that experiments can be fully captured. This entails integrating a variety of tagging and location-tracking technologies so that individual samples can be tracked as they are moved, mixed, heated, centrifuged, etc.. Embedded web-servers can be used to connect laboratory instrumentation to the web so that devices can be controlled and configurations recorded automatically.

With this level of capture, it should be possible to record everything that goes on in the laboratory, obviating the need for imprecise, error-prone, and incomplete notebooks. Furthermore, the applications of the collected data are immense. Not only do they provide a record of all experiments for all time, but also it is now possible to conceive of automated lab tutors that can guide a person by playing back the details of a previous experimental procedure. Scientists can be aided in keeping track of many simultaneous experiments, relieving their cognitive burden and leading to less errors and more effective use of time in the laboratory. Eventually, we can imagine scientists being able to describe a new experiment and have robots carry it out for them automatically.

HEALTH MONITORING

Ubiquitous sensors and inter-networking will radically change our health care. When a patient needs to be observed, a physician will be able to prescribe a collection of sensors that can be swallowed. Ingested daily, they provide chemical, temperature, and physiological data that is collected by a portable embedded web-server over an RF link. The physician can thus observe how a therapy affects the patient continuously rather than in the discrete samples taken during office visits. In fact, sensors and actuators specific to a particular medication can be used to time-release appropriate doses only when necessary, and alert physicians to side-effects, or unwanted interactions as they occur. One can thus imagine a world in which new drugs are developed along with their monitoring sensors and releasing actuators that will guarantee their safe and effective use.

Privacy concerns abound in such a world. Clearly, a secure web-based architecture for patient medical data will have to be deployed. Patients should be able to keep their complete medical history for themselves at a third-party service. Physicians can then be authorized by the patient to look at subsets of that data. Insurance companies will similarly see legislated subsets. However, the patient is free to purchase third-party services that can augment or backup physician's services. For example, a drug interaction service that relates possible side-effects to the patient, much as a pharmacist may do today, but with more complete information. Commerce applications as well as reminder services abound as well.

CONCLUSION

We are entering a new phase in the development of computing. Embedded processing is becoming powerful enough to tackle an ever-widening range of applications. Wireless and wired networking is becoming ubiquitous, cheap, and low-power so that we can envision interconnecting all our embedded processors. The synergy this will foster will make the much touted digital convergence in the desktop publishing and entertainment industries look like a mere blip in the progress of computing.

There are many challenges in making this revolution improve the quality of our lives. Some of these are technical and involve new approaches to software development and deployment, new networking protocols, the organization of network-based services, and techniques for self-organization, self-configuration, and self-monitoring of large distributed systems. Other challenges are not purely technical. Among these are privacy and security concerns that will have to be addressed across a broad front, including public policy and law. A still greater challenge is the development of new business models that will permit the horizontal interoperability of our devices and services, thus enabling consumer choices and the flexibility we all deserve.

ACKNOWLEDGEMENTS

Kurt Partridge for his major role in implementing PARC's embedded web-server; Jeff Hightower for his implementation of advanced tagging techniques; the Extreme UI group at Xerox PARC for their work on user scenarios and to Mark Weiser whose vision of invisible computing inspired us all.

REFERENCES

1. Analog Devices. Technical Note. "Using iMEMS Accelerometers in Instrumentation Applications http://www.analog.com/industry/iMEMS/library/imems_accl.htm
2. Bluetooth, http://www.bluetooth.com/technology/, SIG, 1999
3. HomeRF Working Group, http://www.homerf.org/, 1999
4. Infrared Data Association (IrDA), http://www.irda.org. 1999
5. Moore. G., "VLSI: Some Fundamental Challenges", IEEE Spectrum, Vol. 16, 1979, p.30
6. Portolano Project at the University of Washington, 1999, http://www.cs.washington.edu/research/portolano/.
7. TIRIS. Tag-it Inlays, http://www.tiris.com, Product Bulletin, Texas Instruments.
8. Waldo J., "Jini architecture overview". Technical report, Sun Microsystems, Inc., Palo Alto, CA, 1998.
9. Want R., A. Hopper, V. Falcao, J. Gibbons The Active Badge Location System. ACM TOIS 10(1), Jan 1992 Pages 91-102
10. Want R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., Ellis, J. R., and Weiser, M. "An Overview of the ParcTab Ubiquitous Computing Experiment". *IEEE Personal Communications*, December 1995, pp. 28-43.
11. Weiser, M. The Computer for the 21st Century. *Scientific America*, *265*(3), 1991, pp. 94-104.
12. World Wide Web Consortium. Extensible markup language (XML), 1998. http://www.wc3.org/XML/.