1 Ravi Jain

rjain@telcordia.com

Bio: Ravi Jain received the Ph. D in Computer Science from the University of Texas at Austin in 1992. Prior to that he worked at Syntrex Inc., SES Inc. and the Schlumberger Laboratory for Computer Science on developing communications and systems software, performance modeling, and parallel programming. Since 1992 he has been a Research Scientist and then a Senior Scientist in Applied Research, Telcordia Technologies. Currently Jain is Director of the Middleware and Mobile Applications Research Group at Telcordia.

Jain's research interests include middleware and APIs for supporting mobile users and for advanced Next Generation Network applications. He is also an expert in the areas of design and analysis of protocols, algorithms, and systems for mobile computing and communications. Jain is also chair of several key working groups in the JAIN and Parlay initiatives, industry standards fora established to develop open programmable APIs for integrated Next Generation Networks.

Jain has numerous publications in his research areas and several issued and pending patents, mostly in the area of wireless and personal communications. He is an area editor for ACM MONET, ACM MC2R and IEEE Wireless Communications and has been Guest Editor of several journal Special Issues as well as served on numerous conference program committees. Jain is a member of the Upsilon Pi Epsilon and Phi Kappa Phi honorary societies, a senior member of IEEE, and a member of the ACM.

Diverse and inter-disciplinary approaches: Rather than narrowly pursuing a single vein of research, Jain has developed and applied approaches and techniques results from a variety of disciplines to inform his research in mobile wireless and open systems. Recent examples include:

- principles of economics for meeting Quality-of-Service requirements in wireless systems
- Code BLUE, an award-winning prototype of an interactive wireless dance club system using Bluetooth technology
- randomized distributed edge-coloring algorithms for scheduling parallel I/O transfers
- dynamic telephone numbering services for user privacy and anonymity
- optimal algorithms for personalized information delivery via mobile agents
- middleware for mobile wireless and location-based services

Grand Challenges in Computer Science and Engineering:

Sustainable Design for Pervasive Computing Systems

Ravi Jain
rjain@telcordia.com

2 Introduction

Computer systems have come a long way from the early single-application mainframe behemoths used by specialists, often scientists and engineers, in closed, protective environments. Today, an incredible variety of computers and applications, spanning the globe and beyond, have deep and far-reaching impacts on all parts of society.

One result of these impacts is that a missing dimension in computer science and engineering will come forward as an important concern: minimizing the environmental impact of computers. The more ubiquitous computers become in society, the larger this concern will become. Just as manufacturers of household goods such as appliances and cosmetics increasingly seek to project and differentiate themselves as "green" (to respond to or avoid government regulations and consumer pressure), so will computer system manufacturers. Just as diverse disciplines such as architecture, mechanical engineering, and materials science have developed principles and techniques for environmentally sensitive and sustainable design, it is likely that computer science and engineering will need to follow suit.

Further, we argue that the environmental impact of computers will be increasingly driven not by hardware, but by software. And finally we argue that one grand challenge -- and opportunity -- of computer science and engineering is to develop the new formal models, algorithms, architectures, languages, design patterns, and applications to enable the design and development of computer systems for minimal environmental impact.

In this brief paper we can only sketch the issues and challenges; some additional background and supporting statistics can be found in [Jain02].

3 Pervasive computing

A well-recognized and oft-heralded "grand challenge" of computer science and engineering (CSE) is to make the vision of pervasive or ubiquitous computing a reality, leveraging the dramatic advances in device miniaturization, nanotechnology, and wireless communications.

Pervasive computing is driven by ambitious, exciting and noble goals – to put computing machines in the service of humanity, instead of the other way round; to make computing as useful and unobtrusive as utilities like electricity and water; to produce "calm" rather than distraction; and to bring the benefits of computers to everyone by developing not only powerful, costly machines but "tiny inexpensive ones" [Weiser93, Weiser96]. While this vision has not yet been fully realized, much progress has been made, and researchers and engineers around the world are chipping away at the obstacles. We take it as given that this challenge will, at least to a large degree, be met.

However, little or no thought has been given to the physical final end result of pervasive computing: devices of varying size, weight and complexity, that are useless, obsolete, malfunctioning, or simply broken – in other words, that are garbage. Further, these devices, by their very design and function, are ubiquitous, massively distributed, and embedded in numerous everyday objects and the environment. From digital jewelry and clothing, to networked appliances, sensor networks, smart floors and cyberhomes, pervasive computing offers us not only a glittering future of convenience, comfort and connectivity, but possibly also a legacy of deadening clutter and dangerous trash: plastics that do not biodegrade, heavy metals that are carcinogenic, gases from production and incineration that are toxic, and landfills that threaten generations to come.

To gauge this possibility, consider current – i.e., non-pervasive – computing. It has been estimated that over three-quarters of all computers ever bought in the U.S. are stored in people's attics, basements, office closets and pantries [MCC96]. By the year 2004, experts estimate there will be over 315 million obsolete computers in the U.S., the vast majority destined for landfills, incinerators or hazardous waste exports [NSC99]. Consider that computer equipment is a complicated assembly of over a thousand materials, of which many, such as lead, cadmium and mercury, are known to be highly toxic; what may be worse, the effects of some of these are still unknown. Leaving physical waste aside, computing, telephony and networking equipment now accounts for a significant fraction of the total energy consumption in the U.S. [EStar]. While mobile computing devices such as laptops individually are more energy-efficient than before, the overall energy consumption due to such devices continues to increase [Paradiso00]. Thus the environmental impacts of computers have been a subject of growing concern over the past decade [DoE].

Generally most pervasive computing devices – although by no means all -- will have one significant environmental advantage over traditional computers: that they are physically smaller and inherently consume less material. However, they have other disadvantages. They will be far more numerous; low cost will encourage rapid replacement; less mature technology will become obsolete faster[1]; disposable versions of some devices, like disposable cell phones [HopOn01, Telespree01], will soon emerge; they will tend to use batteries, which often contain heavy metals and are an environmental hazard in themselves, rather than fixed AC power. In addition, their small size, weight, embedding in other materials and overall design for ubiquity will disperse them widely, making them more likely to be lost, forgotten, or simply abandoned, and making proper collection, recycling or disposal harder. Finally, pervasive computing devices, to meet their goal, will be truly global, bringing computer environmental impacts to regions of the world where little or none exist at present.

4 Computer science & engineering implications

Our intent is not to present an alarmist view of global environmental collapse due to computers, but to argue that CSE faces interesting challenges and opportunities for reducing the environmental risks of pervasive computing by innovation.

We believe that effective reduction of resource consumption, reuse of resources, and recycling of materials – the reduce/reuse/recycle mantra -- needs to be an integral part of the CSE design process, not an afterthought or enhancement. This holds for manufactured products in general, e.g.: "Very few objects of modern consumption were designed with recycling in mind. If the process is truly to save money and materials, products must be designed from the very beginning to be recycled …" [McB98]. In fact, unlike current computing, where environmental concerns were only raised after the proliferation of computers, pervasive computing offers us a unique opportunity to apply the environmental consciousness raised by current technology *while we are still at the start* of the next wave of technology proliferation.

Of course, much of pervasive computing research focuses on design to minimize resource consumption, mainly because of limitations of device size, weight and capabilities that make the research interesting in the first place. However, the cost of resources does not take into account environmental impacts that are external to the producer, i.e., after a product leaves the factory[2]. Thus CSE research should consider minimizing not only production and operation costs but total lifecycle impacts, i.e., choosing techniques to reduce the costs of reuse, recycling, disposal and incineration. This complicates the design process, and also makes it more challenging: developing new models, metrics and techniques to carry out or evaluate such designs is a fundamental and worthy goal.

5 Software challenges

Pervasive computing differs from traditional science and engineering disciplines in that increasingly it is software that determines hardware resource consumption and residue rather than hardware itself. We briefly mention some issues here.

Software sprawl is a key contributor to needless hardware consumption. Each generation of operating system and application software is larger and more complex, requiring hardware upgrades and rapidly obsolescing entire systems even though most users actually utilize only a fraction of the new capabilities. Thus, for example, CSE needs to develop new languages, programming interfaces and design techniques for dynamically discovering device and system capabilities and user requirements; automatically and securely downloading, upgrading and configuring software components (at all system levels from software radios to application modules); and billing, security and system mechanisms to support business models such as software leasing and pay-per-use. *Data sprawl* is another factor since most systems, while minimizing storage "in the small", store data inefficiently "in the large". As two small instances, most word processors store the complete document for each revision rather than just the revisions; inter-office email attachments (which may soon include not only images but multimedia) are copied to every mailbox instead of being automatically stripped out and sent as a pointer to a single copy. The list of such inefficiencies can easily be made longer, mainly because the "storage is cheap"

[1] For example, one year's device may be replaced because it is not WAP-enabled, and the following year the replacement may be replaced because it is not Bluetooth-enabled.

[2] This is especially true in the U.S., although recently in Europe "extended producer responsibility" policies are shifting the burden of waste electrical and electronic equipment to the manufacturer [EC00].

refrain hides the human and environmental costs of storing, retrieving and managing useless and outdated information. New techniques for self-managing and self-destroying data, under user policy control, are among techniques that can help. *Inadequate formal models* of computing, while highly sophisticated for modeling reasoning about correctness as well as asymptotic time and storage complexity, are in their infancy when it comes to scalability analysis of energy consumption or reasoning about software reusability and assembly. New theoretical frameworks and techniques, grounded in firm empirical usefulness, are required to guide system and algorithm development for pervasive computing. *Myopic system engineering* that tends to focus on production cost, needs to expand to develop software tools and hardware/software codesign techniques to minimize total lifecycle costs.

In short, we believe that sustainable design for pervasive computing systems will bring a new and much-needed dimension to the entire discipline of computer science and engineering, and its scope and complexity forms a grand-challenge for the field.

References

[EC00] European Commission, Commission tackles growing problem of electrical and electronic waste, DN: IP/00/602, Press Release, June 13, 2000. See http://europa.eu.int/rapid/start/cgi/guesten.ksh?p_action.gettxt=gt&doc=IP/00/602|0|RAPID&lg=EN

[DoE] U.S. Dept of Energy, The Energy Star Program, See http://www.eren.doe.gov/cities_counties/saving1.html.

[Estar] U. S. Environment Protection Agency, Press Releases: "EPA Administrator Looks to Telecommunications Industry for Increased Energy Efficiency Opportunities", 11/16/2001; "Energy Star to Launch Label for Telephony Products at Consumer Electronics Show in Las Vegas, January 8-11, 2002."

[HopOn01] Hop -On Communications, See http:// www.hop-onwireless.com

[Jain02] R. Jain, Rethinking pervasive computing, NSF Workshop on Context-Aware Mobile and Pervasive Data Management, Jan. 2002. See http://www.cs.brown.edu/nsfmobile/wshop.html

[McB98] W. McDonough and M. Braungart, The NEXT Industrial Revolution, *Atlantic Monthly*, October 1998.

[MCC96] Electronics Industry Roadmap, Microelectronics and Computer Technology Corporation, Austin, TX, 1996.

[NSC99] Electronic product recovery and recycling baseline report, National Safety Council, Washington, DC, 1999.

[Paradiso00] J. Paradiso, Renewable energy sources for the future of mobile and embedded computing, MIT Media Lab, Mar. 2000.

[Telespree01] Telespree, Inc, http://www.telespree.com

[Weiser93] M. Weiser, Some Computer Science Problems in Ubiquitous Computing, Communications of the ACM, July 1993.

[Wesier96] M. Weiser and J. S. Brown, The coming age of calm technology, Xerox PARC, Oct. 1996. See http://www.ubiq.com/hypertext/weiser/acmfuture2endnote.htm