THE TRILLION-NODE NETWORK

Peter Lucas
Founder, Principal, Board Chair
MAYA Design, Inc.

Published March 1999

ABSTRACT

It is widely accepted that in the foreseeable future the worldwide network of computing devices will grow to billions, or even tens of billions of nodes. However, if we broaden our consideration to include networks of information devices (all artificial systems that deal in any way with information), then we are likely to be faced with much larger numbers. A network of one trillion devices is not inconceivable. Design at this scale cannot rely on engineering discipline alone. It will entail the kind of loose consensus among communities of designers that, in traditional architecture and design, goes under the name of style.”

Keywords

Distributed computing, information architecture, networking, information design

A MAYA® paper

MAYA Design, Inc.
2730 Sidney Street
Pittsburgh, PA 15203

T: 412-488-2900
F: 412-488-2940
maya@maya.com
www.maya.com

INTRODUCTION

There is a growing consensus that we are on the cusp of a discontinuity in the evolution of computing centered on the emergence of radically distributed, network-centric systems. One increasingly encounters statements to the effect that we are approaching a "paradigm shift unmatched since the advent of the personal computer." Although the basic soundness of these predictions seems irrefutable, the soothsayers can be divided into two camps when it comes to the details of their predictions.

One camp—the "One Huge Computer" (OHC) school—sees Java as the last brick in the foundation of a system that will finally liberate computation from island like uniprocessor computers into the wide world of the Net. By this vision, the personal computer will be deconstructed into functionally specialized component computers. Every disk drive will become a network file server, every general purpose CPU a compute server, and every I/O device a data source or sink. Glued together via some simple, general mechanism for network auto configuration such as Sun's Jini [4] architecture, the network itself will begin to behave as one vastly distributed, constantly evolving multicomputer.

Members of the other camp—the "Information Appliance" (IA) school—look up from their Palm Pilots and see a world rapidly filling up with wildly diverse, small, cheap, semi-autonomous products, each having at least some ability to process information. With the marginal cost of adding a bit of computational ability to manufactured products quickly becoming negligible, "smart products" are becoming the rule rather than the exception. In addition to the obvious cases of personal digital assistants, cell phones, automobiles, wristwatches and so on, other real-life examples include a bowling ball that will monitor and critique its user's form, swimming goggles that will accumulate statistics on its wearer's exercise regimen, and a birthday candle that plays an electronic rendition of "Happy Birthday" when lighted. Further, low-cost short-haul communication standards such as IRDA using infrared and Bluetooth for RF will soon make it feasible for even the most humble of such devices to possess the ability to communicate with their peers.

These two visions of the future do not contradict each other, but they do have different emphases and they raise different issues. It is the thesis of this paper that, not only must we take both scenarios seriously, but there are issues that become apparent only when both trends are contemplated simultaneously. Moreover, some of these issues are of a kind that will never be successfully addressed by engineering principles alone. Rather, they will require a kind of creative collaboration and shared consensus among communities of design professionals which, up until now, has more typically characterized such traditional communities of design practice as architecture and industrial design than HCI or software engineering.

INFORMATION DEVICES

You may have assumed that the title of this paper is an attempt at hyperbole, but it is not—I mean to be taken literally. The so-called "next generation Internet" project is essentially about bandwidth. Surely, however, the "Internet-after-next" will be about scalability. Even if we limit ourselves to the OHC agenda alone, we are faced with non-trivial issues of scalability. By one estimate, the number of human users of the Internet will reach one billion by the year 2005 [8]. In the context of such growth, it is clear that building "one huge computer" would imply an Internet consisting of multiple billions of processors.

These numbers, although challenging, are within the reach of more-or-less conventional approaches to network architecture. But the kinds of scalability implicated by the IA agenda are another matter entirely. There were more than two billion microprocessors manufactured in 1996[12] alone. This statistic implies that in all likelihood there are now more processors on the planet than there are people. The overwhelming majority of these processors have gone not into anything we would think of as a general purpose computer, but rather into cars and cellphones and PDAs … and bowling balls and swim goggles and birthday candles (all right, the chip in the candle probably wasn't actually a microprocessor, but as I will argue, this is beside the point). If we were to aspire to design a new network architecture meant to internetwork all of these processors, then the adoption of an approach that would not scale to one trillion nodes would represent shortsightedness on a level not seen since the adoption of the two-digit date.

But, does such an aspiration make any sense? No one is going to want to put bowling balls on the Internet. Is the notion of a Trillion-Node Network of any practical interest? If we are merely talking of conventional networks of computers (even radically deconstructed computers), then the answer is "no." Liberal estimates of the need for such machines might yield tens- or perhaps hundreds- of billions of machines, but an order of magnitude beyond that is hard to imagine.

However, the IA agenda isn't really about computers per se. To discuss what it is about requires a term superordinate to "computer" that also includes other devices whose functions involve operations on information. I propose the term "information device" (or "infotron" for short) for this purpose, an infotron is "a device whose intended function includes the capture, storage, retrieval, transmission, display, or processing of information." One might argue that "information device" is just a pedantic synonym for "computer," but this is not the case.

First of all, there have been information devices far longer than there have been computers: The whistle of a steam engine is an information device, as, for that matter, are pen and ink. Moreover, even many modern information devices do not actually compute, or do so only in ways that are incidental to their intended function.

It is a bit surprising that no term equivalent to "infotron" is in common usage. The reason, I think, has to do with the fact that the concept of "information" in its modern sense is of rather recent origin. It was only in 1948 that Claude Shannon provided a rigorous framework within which to think about the concept of information [9]. (Indeed, it is interesting to speculate about exactly what the word "information" connoted in its pre-computer usage. I suspect that it was much more a vague descriptive word than a technical term.) Conversely, since the introduction of the computer, that machine has loomed so large as the canonical information device that it is easy to forget that there are and have been others.

It is important to take seriously the "pen and ink" example given above: Not all infotrons are electronic. If the reader has trouble taking a printed page seriously as an information device, then consider a printed bar code. If this is still not compelling, then how about a CD/ROM disk? Where should the line be drawn? Each of these examples encodes information optically—if one accepts the CD/ROM as an infotron, then I would argue that one should accept them all. The point is that, although ability to perform computation may be a requirement to be considered a computing device, this ability is not necessary to qualify as an information device. But what has all of this to do with networking? We build networks of computers, but we can't speak of networks of CD/ROMs. Or can we? Couldn't a CD/ROM drive be thought of as simply a network adapter for disks? That is, isn't a CD/ROM disk mounted in a properly configured server meaningfully "on" the Net? And if so, is there any fundamental difference between a CD/ROM in a drive and a printed page in a scanner? Couldn't we think of fax machines as simply devices for "connecting" two pieces of paper for the purpose of transporting information from one to the other?

My point in pursuing this somewhat strained line of rhetoric is to drive home the point that a network of information devices is not at all the same animal as a network of computers. In particular, the focus shifts away from the computing and communication machinery that makes the network work, and toward the flows of information that courses through that network. The devices themselves merely constitute the physical substrate of a radically distributed, undesigned, unadministered worldwide dataflow machine. The challenge is to conceive of an architecture that will scale to trillions of devices that are capable (in general) of only local communication, with no central registration authority, that together will support the free, liquid flow of information objects wherever the currents of human activity take them. In this vision, the devices are vessels and channels for the information—the data flow through them.

It is the major theme of this paper that the design of these flows is an activity that differs qualitatively from the design of computer hardware and software, and that this is an activity that requires a unique collaboration between engineers and other designers—specifically information designers. Moreover, if we are talking about a network of information devices, rather than a network of computers (assuming that "network" is still the appropriate term), then the idea of a trillion nodes is not at all preposterous.

THE GRAND CHALLENGES

One trillion is a big number. There are few precedents for artificial systems of any kind that contain a trillion components. Design on this scale obviously poses many unique challenges to the designer. Such systems cannot be designed component-by-component, or even subsystem-by subsystem. The best the designer can do is to understand the principal challenges to the integrity of the system and to attempt to guide the emergence of the system in ways that address these challenges. In this domain, three challenges emerge as preeminent. Those are the need for scalability, for tractability, and for comprehensiveness.

Scalability

For all practical purposes, the requirement of the Trillion-Node Network is for unbounded scalability. This is a severe requirement. On the one hand, there is a clear need for some kind of ubiquitous standardization on a grand scale. On the other hand, the need for unbounded scalability places stringent restrictions on the use of central authorities of any kind. Any introduction of central address registration authorities, semantically coordinated global name spaces, universal ontologies, etc., represents costs and potential bottlenecks that cannot, in general, be tolerated. Any organization scheme that requires each device to receive any individual attention whatsoever in order to join the network is simply precluded—you just can't afford it. Rather, the designer must assume what might be called a "deist" design philosophy. That is, the designer must adopt the role as being the creator of an evolutionary framework—the "laws of physics" if you will—for a sub-universe which will unfold on its own, driven by local decisions and environmental pressures, not by the active supervision of any god-like supervisor.

One of the few existing human-created systems of similar aggregate complexity is the worldwide economy. History has demonstrated that comprehensive central planning does not work in an economy, and there is every reason to believe that it will not work in vast networks, either. In both cases, there is an essential role for universal standards. But this role is extremely narrow and of a particular type. Specifically, successful universal standards tend to be syntactic rather than semantic. Their role is to enforce only enough standardization to support the existence of relatively efficient markets. Thus, for example, governments establish universal currencies in order to define a common medium of exchange. Just so, the Trillion-Node Network will not become a reality until there is an agreement on a common “currency” to serve as the universal medium of exchange of information. This amounts to the establishment of a universal “information architecture”—a topic to which we shall return. Attempts to mandate universal standards for the semantics of transactions, on the other hand, are doomed to fail. Rather, these should be permitted to evolve bottom up—emerging by natural selection out of the cauldron of market activity. In our economy, standard contract terms are not codified by any controlling authority—they have evolved over time, reflecting the accumulated wisdom of many billions of individual transactions. Establishing conditions that support and encourage analogous evolution should be our highest goal as we design future information systems.

Tractability

Our second grand challenge is that our system remains tractable. That is, not only must it be able to grow arbitrarily; it must be able to do so without having any critical aspect of the system become unmanageable. The biggest single threat to the manageability of large systems is, of course, that they tend to become intractably complex. In particular, an important threshold exists at which an individual designer is no longer capable of fully understanding the details of a system. This threshold is important because it is the point at which individual skill can no longer be depended upon as the source of integrity of a design. Once this point is reached, the only real recourse is to rely on formal management methodologies as a substitute for individual design skill. But, as systems tend toward the very complex, requisite management efforts tend toward the heroic. At the extreme edge of tractability, we might occasionally succeed at producing a space shuttle or a Windows 98, but it is not obvious that we have the skill to push much further into the uncharted frontiers of complexity. It is unlikely that design management alone will get us anywhere near a fully engineered Trillion-Node Network.

The other primary threat to tractability is also complexity related: If increasing the size of a system implies any significant increase in the complexity that the system exposes to its users, then the system cannot scale indefinitely. For such a system, arbitrary growth means arbitrary complexity. Eventually, the bounds of human capability will be reached, and successful growth will stop.

The antidote to these complexity-related threats to tractability is well known: complex systems must be modular and hierarchically decomposable. As Herb Simon argues in his famous paper The Architecture of Complexity, [10] decomposable systems are the rule in nature for precisely this reason, and there is every reason to believe that complex artificial systems should be designed this way as well.

Comprehensiveness

The last of our grand challenges is the requirement for comprehensiveness. The Trillion-Node Network amounts to an agenda for interoperability on a grand scale. We are talking not just of One Huge Computer, but One Huge Dataflow, encompassing devices representing the full diversity of human artifice, from the key fob that sends messages to your car to unlock its doors, all the way to supercomputers, and everything in between. What can we say about the design of key fobs that will have any relevance whatsoever to the design of supercomputers? More generally, what possible design principles could we lay down that would contribute to our dream of supporting the free flow of information among a trillion devices of a billion different designs and intended uses?

Before answering this question, let me state some things that probably won’t work. First of all, coordinated design won’t work—not at this scale. The world of the key fob designer has so little in common with the world of the supercomputer designer as to render any expectation of collaboration just silly. There is neither economic motivation nor practical possibility for these communities of practice to cooperate in any interesting way. Multiplied a billionfold to cover the breadth of our aspirations, the situation is clearly hopeless. Nor will standards save us. Even within a single community of practice, the standards process is fraught with conflict and resultant complexity. In an attempt to resolve conflicting interests, the parties to standards deliberations inevitably resort to compromise by-superset, yielding standards documents that achieve consensus at the expense of simplicity and elegance. Although this process often yields great value in local situations, applied across the vast span of information devices, the standards process is not tenable. Bowling balls will not run Java—and if I am wrong about that, then we can move down to birthday candles. The thought that universal standards adequate to support universal interoperability will ever emerge is a pipe dream.

INFORMATION ARCHITECTURE AND DESIGN “STYLE”

If "Grand Design" won't work, what will? The answer, I think can be found in the traditional notion of "design style." Note that I do not mean "styling" in the sense of Cadillac tail fins. Rather, I mean "Style" in the sense of "Baroque," "Gothic," or "Postmodern." As Walter Darwin Teague put it, at times when there is a dominant style then

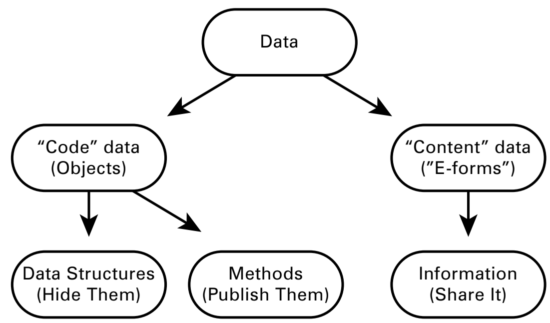
> … a single character of design gets itself expressed in whatever is made at the time, and not a chair, a teapot, a necklace or a summerhouse comes into existence except in a form which harmonizes with everything else being made at that time…. The scene has unity, harmony, repose, and at least one irritant is absent from the social organism.[11]

"Style" in this sense represents the middle distance between the rigor and completeness of engineering design and the free-form expressiveness of individual creativity. It is neither the Corbussian fantasy of a completely designed "radiant city" nor the free-for-all of an unplanned commercial strip, but rather something in between—vague enough so as not to constrain progress and individual creativity, but specific enough to impart a sense of harmony onto ensembles of artifacts created under its influence. If one were to telephone a furniture store and—sight unseen—order a room full of, say, Mission Style furniture, the result may not merit coverage in Architectural Digest, but it is likely to hang together pretty well.

Where do styles come from? Well, they don't come from committees, and (at least in general) they don't come from engineers. Rather, they emerge as rough shared consensus among communities of practice—more specifically among communities of designers. This is unfamiliar territory to many engineering-oriented designers, but it represents the principal point of this paper. When designing at the scale with which we are now faced, we will inevitably be forced to abandon our dreams of complete rigor, and when we do, the only remaining alternative to chaos is the loose but pervasive consensual shared agenda that we refer to as "style."

Actually, styles are not altogether absent from the computing scene. System architects have evolved a very definite style for the building of computers themselves. The packaging of logic in functionally specialized ICs; putting main memory chips on little daughterboards; the use of APIs; object-orientation; and semi-standardized datatypes—all of these are "elements of style" within the engineering community. Similarly, within the CHI community, the WIMP paradigm represents a loose, evolving but near-universal style of user interface design. But the Trillion-Node Network will require the emergence of a third distinct kind of style, namely a style of information architecture. Lying just above systems architecture (which deals with how the information devices themselves are built) and just below UI architecture (which is about how systems are presented to users), information architecture deals with the design of the information itself. As I have argued, the Trillion-Node Network should be thought of as a vast, incredibly heterogeneous worldwide dataflow of information. The only thing in common across all of this vastness is the information itself, and it is here that we must concentrate new design effort if we are to achieve a semblance of global integrity.

Figure 1 Data Architecture



The notion of information architecture deserves a bit of elaboration. It is analogous to, but distinct from the kind of system architecture represented by object-oriented design. Both are instances of the broader concept of "data architectures." The relationship between the two may be depicted as above.

As this figure suggests, data may be partitioned into two categories: "code" data and "content" data. In a basic sense, the "style" that goes under the name "object orientation" (OO) is "about" code data. OO was conceived, and has found success, as a style for engineering computer software. Its basic tenants include strict encapsulation of internal mechanisms (including internal data structures), the externalization of behaviors in the form of published "methods," and object specialization via inheritance. These are profoundly wise principles of system design. They are, however, nearly nonsensical as principles of information design. As we have seen, many information devices do not compute, and many more compute only incidentally to their design purpose. If this is so, what could it possibly mean to "hide the data" of entities that consist only of data? Can we really afford to insist that every information object, no matter how humble, be required to carry with it enough computing power to implement methods sufficient to support a data encapsulation scheme? This amounts to a requirement that all chunks of information travel everywhere as "active objects" and that there be sufficient uniformity throughout the computing landscape to make such a scheme feasible.

Such requirements are clearly absurd, and the designers of the OO style did not intend them. Nonetheless, absent an analogous style for the design of information, many developers have made misguided attempts to blindly apply OO software design principles to information design problems. Thus, for example, CORBA, conceived as an architecture for creating distributed, object-oriented computing systems—is being misapplied as a framework for developing massive ontologies of declarative (that is non-computational) data. Such efforts are doomed to fail. Although they would be a very powerful way to achieve tractability, they are too cumbersome to be scalable and—more importantly—by not recognizing the wide diversity of information devices, they will fall far short of achieving the comprehensiveness essential to the Trillion-Node Network. What is needed instead, as suggested in Figure 1, is a parallel style of design for information objects. There has been remarkably little recent work in this area. Not since the development of the relational model for database design and the development of SGML, both several decades old, have serious new informational architectural efforts been mounted (in making this statement, I am considering the recent XML initiative as a much-needed revisiting of the SGML agenda).

ELEMENTS OF STYLE IN INFORMATION ARCHITECTURE

What form would a universal style of information design take? If in fact the rigors of data encapsulation and behavior inheritance are unsuitable as an information architecture, what can we substitute as a source of tractability? During the past decade, designers at our studio have explored this issue deeply within the context of numerous commercial information-design efforts as well as two very large-scale research projects—first Workscape [5], [1] (an experimental office document management system developed under contract to Digital Equipment Corporation in the early 1990s) and later, Visage [3], [6] (an ongoing information exploration project funded primarily by DARPA and the Army Research Laboratory). Although Workscape is best known for its 3-dimensional user-interface paradigm, and Visage for its information-centric model, they share a common style of information architecture that has elements in common with work emerging from a number of other laboratories. The core of this common style is a design principle that Clifford Neuman has, in the context of his work of the Prospero distributed file system [7], labeled "layered semantics. "The essence of this style of information design is that information objects should be built up from successive representational layers, with the lower layers consisting of simple, universal common syntactic forms, with semantically specific representational features being limited to less-universal higher levels.

The notion of layered semantics is as obvious as it is rarely achieved. As Michael Dertouzos has put it: "Achieving some basic degree of understanding among different computers to make automatization possible is not as technically difficult as it sounds. However, it does require one very difficult commodity: human consensus." [2]. In fact, I will argue that, through all the history of electronic information processing, we have succeeded exactly twice at establishing near-universal consensus on elements of information architecture. The first of these was in the 1950s when, after years of experimentation with analog computers and with decimal computers, the bit was once-and-for-all established as the universal first layer of data representation. It took twenty years, until the 1970's, to reach the second great consensus: the near universal adoption of the 8-bit byte as the second layer of representation. The importance of these two basic standards in supporting interoperability and data liquidity across computing devices cannot be overstated. They form the basis of standard integrated circuits, standard disk drives, communications protocols… they literally pervade all of computing.

Forty years after the bit and twenty years after the byte, what is the likely candidate for the next universally adopted layer? CORBA? The Java virtual machine? I suspect that next step will be more modest. A plausible candidate is the attribute/value pair. Dertouzos has long advocated simple named values—which he calls "E-forms" as the common basis for the "information marketplace. "Similarly, Neuman's Prospero system uses uniform attribute-value pairs as the data "containers" used to build higher level mechanisms such as the Archie Internet search service. In our own work, both Workscape and Visage achieve their architectural integrity from the disciplined layering of all mechanisms upon attribute/value pairs.

What commends attribute/value pairs as the next universal "element of style" in information architecture? The answer is reflected in Dertouzos's choice of the "marketplace" metaphor in describing his vision of a future of information liquidity. E-forms are necessary and sufficient to form the currency of a new marketplace of design ideas and mechanisms sufficient to support the evolution of the Trillion-Node Network. They are necessary in that they are the simplest possible increment in semantics beyond the byte that is likely to support a significant increment of universal standardization across the diverse span of infotrons. They are sufficient in that they are certainly adequate to define a class of universal service that can be adapted to nearly any information interchange task, and also in that they provide a sufficiently well-defined structure to permit the engineering of a large class of new data storage and transport devices and standards independent of higher-level semantic content. To draw an analogy with the OO methodology, bundles of attribute/value pairs form the "objects" of the system, and layered semantics substitutes for inheritance as the source of decomposability.

Standards such as this represent "couplers" that permit interoperability among components while preserving the ability for them to continue to evolve separately—each according to the pressures of their separate markets. Bits and bytes have been the couplers, for example, between the evolution of modems and the evolution of the Internet. Modem performance has come so far so quickly precisely because a modem's task reduces to the simply stated goal of moving bytes to and fro without regard to their content. (I am sure that some day someone will get the bright idea of building a web-aware modem. "Think of the extra compression that could be achieved if the modem had knowledge of the structure of HTML," they will say. But such an approach would be disastrous because it would forever link the evolution of modems to that of HTML pages, thus hampering the development of both.) Similarly, as Dertouzos has argued forcefully, E-forms will form the foundation for the development of a wide range of consensual standardization of simple transactions in support of diverse areas of human activity. Anything less will do no good. Anything more is unlikely to achieve the kind of universal acceptance that is needed.

Given the somewhat grandiose starting point of this essay, it may seem that we have traveled awfully far only to arrive at so mundane a destination as attribute/value pairs. It should be understood that this is just one example of a number of emerging stylistic elements that together will help us to achieve "a single character of design" across all that we build today. By themselves, each of these elements will seem mundane. But, that is the way of styles. They are a far cry from engineering plans. They are mere scaffolds for the creative work of thousands of independent designers—not in themselves profound. But if they are well-chosen, and if they are nurtured by a community of designers, engineers, and artists who think like designers before they begin to think like engineers or like artists, then the results can be transformational. Modernism transformed the face of the urban landscape, as did Postmodernism after it. The bit and the byte transformed the face of computing. The Trillion-Node Network will not be designed from the top down, but nor will it emerge entirely on its own. Its evolution will be utterly dependent on the subtle but pervasive effects of a shared consensual style of information architecture. Such a style is not "about" hardware or software or user interfaces. Rather, it is about an emerging ecology of people, information, and devices. This is an agenda for design in its deepest sense. If these arguments are valid, then the emergence of the Trillion-Node Network will inevitably coincide with the emergence of the first mature community of information designers.

ACKNOWLEDGEMENTS

Portions of this work were funded by DARPA contract #F30602-97-C-0262.

The ideas presented here have emerged from long collaboration with my colleagues at MAYA, and with our clients. I must particularly thank Jeff Senn, whose contributions to these ideas cannot be fully separated from my own, and Susan Salis, for whose editorial assistance my readers should feel grateful.

REFERENCES

1. Ballay, J.M. Designing Workscape: An Interdisciplinary Experience. Proceedings of CHI '94(April 1994). ACM Press, 10-15.
2. Dertouzos, Michael. What Will Be. Harper Collins,New York, NY, 1997.
3. Gomberg, Cristina, Lucas, Peter, and Roth, Steven. Visage: Dynamic Information Exploration. Proceedings of CHI 96 (Denver CO, April 1996). ACM Press.
4. Kelly, K. & Reis, S. One Huge Computer. Wired (August 1988). 129-135, 170.
5. Lucas, Peter. Workscape Document Management System. Proceedings of CHI '94 (April 1994). ACM Press.
6. Lucas, Peter, and Roth, Steven. Exploring Information with Visage. Proceedings of CHI 96 (April 1996).ACM Press.
7. Neuman, Clifford. The Prospero File System: A global file system based on the Virtual System Model. Computing Systems 5 (4), (Fall 1992), 407-432.
8. Report Urges U.S. to Take the Long View. Science V.281, (August 1998), 1125.
9. Shannon, Claude E. (1948) A Mathematical Theory of Information, Bell System Technical Journal 27, 379-423, 623-656.
10. Simon, H. The Architecture of Complexity. in The Sciences of the Artificial. Third Edition. (Cambridge, MA, 1996). MIT Press, 183-216.
11. Teague, Walter Darwin Design This Day: The Technique of Order in the Machine Age (New York, 1940). Harcourt, Brace and Company, 207.
12. Walsh, Patrick J. The Hidden World of Real-time Operating Systems. Portable Design (2)5 (May 1997), 49-50.