Viewpoint | Harold W. Lawson

Rebirth of the Computer Industry

What types of products are required and who will provide them?

Last year, Paul Horn, the IBM vice president of research, made an important announcement concerning autonomous computing [1]. Horn confessed the complexity sins of the computer industry and proposed an ambitious program for cleansing the sins. More recently, Bill Gates, in a Microsoft internal memorandum, implicitly admited of past complexity sins and introduced plans to redirect development efforts toward providing trustworthy systems. Finally. I have been waiting 36 years for this development.

Has the rebirth of the computer industry begun? Let's hope so. Is the end of the Microsoft-Intel era approaching? It is likely this will occur during the next few years. There are a number of signals pointing in this direction, including: the general state of the IT industry; awareness of enormous risks associated with the Internet; U.S. and European antitrust litigation; the open source movement; the IBM autonomous computing announcement; redirection of Microsoft product development; competition from the telecommunications industry; and resistance in investing in new general-purpose computer products and services. Given these signals, it is difficult to see how Wintel (Windows-Intel) can lift itself from the current situation by continuing to deliver unnecessarily complex insecure platforms of hardware and software to a broad marketplace. The world is, and should be, awaiting the arrival of significantly improved computer industry products.

What types of new products are required, and who will provide them? It is contended that the most bene-ficial path involves a rebirth of the computer industry facilitating competition between a plurality of global actors who become involved in producing and marketing quality products and services. However, there are other potential scenarios that can lead to the end of current Wintel platform dominance.

To understand the immature state of the general-purpose computer industry, it is important to review the industry history (1947–2002) divided into three eras of approximately 18 to 19 years each.

Adolescence (1947–1965)

The industry was born in 1947 when the Eckert-Mauchly Computer Corporation produced the first commercial computer systems. While IBM attained a strong foothold by the mid-1950s, there was significant competition from Univac, RCA, General Electric, Burroughs, Bendix, Control Data, Honeywell, Philco, and Sylvania. In Europe, companies like English Electric, Marconi, Bull, and Siemens competed. In the Soviet Union and Japan there were beginnings of computer industries.

A variety of hardware architectures evolved and each supplier developed its own system software, utility programs, and services. There was a tendency to separate scientific and technical computing from administrative computing by providing separate platform types. The needs of military and space programs, particularly in the U.S. and the Soviet Union, led to significant spending and technological advances in the industry. In 1960, with strong U.S. Government support, the industry developed its first important verifiable computer industry standard; namely, Cobol. After an active competitive environ-

ment in the 1950s and early 1960s, the industry consolidated, resulting in the emergence of monopolistic actors in the following two eras.

IBM Era (1965–1983)

Driven by the belief that a single-platform architecture for almost all forms of computing was essential, IBM developed the System/360 line of computers and its system software; namely Operating System/360 with supporting compilers, utilities, and services. With the results in hand, it can be constituted that this altruistic vision turned out to become the beginnings of a march toward the "black hole of complexity" (verified by the IBM announcement by Paul Horn). The common 360 hardware architecture, being a compromise, was a poor basis for all types of computing, thus requiring significant quantities of code to accomplish computing tasks. Further, the demands placed upon OS/360 led to an unprecedentedly complex suite of system software, developed by a cast of thousands, that nobody completely understood. Every release of the operating system both corrected and introduced new software "bugs."

Despite poor product quality, the extremely strong IBM marketing organization more or less forced 360 upon the world. A side effect of the significant quantity of "unnecessary complexity" in IBM products was the birth of many spin-off companies. These IBM followers introduced better versions of parts to the IBM system software products, provided consulting help to 360 customers that did not have a chance of mastering 360 complexity, and made fortunes on related education and training. There is definitely lots of money to be made in complexity.

Wintel Era (1983–Present)

The IBM plans to continue its market dominance into the PC era backfired and Wintel moved toward dominance. The Intel hardware architecture evolved from the early 1970 days of 4004, 8008, 8080, x86, to the Pentium line. During the 1970s, the Intel products were deployed in rather simple systems and provided a usable base hardware technology. As the Wintel era unfolded, the same primitive hardware architecture was evolved in rapid steps to become the basis for the extremely complex system software products developed by Microsoft. Thus, a spiral developed: more complex software led to a demand for greater hardware performance and storage capacity, which led to more hardware performance and available storage capacity, which led to even more complex software often filled with unnecessary functionality.

The insanity in the hardware-software spiral that has led the world deeper and deeper into the black hole of complexity should be self-evident. The Wintel complexity has had an even stronger side effect than in the IBM era, with Wintel followers making fortunes as complexity ombudsmen in the form of consultants, educators, and trainers.

On the plus side, inexpensive hardware and useful application products have arisen including email programs, Web browsers, word processors, presentation software, spreadsheets, databases, modeling, and simulation. Furthermore, an enormous market evolved for computer games. The almost mass hysteria around these products caused users to accept the poor platform quality. The costs and frustrations of rebooting, loss of critical information, poor security leading to hacker attacks, criminal acts, and so on, are accepted as common phenomena. Absolutely astounding!

Is this era of Wintel platform dominance coming to an end? Most likely. The big question is: What will follow?

Moving to Maturity

What is needed to move the computer industry from its immature state to maturity? It will not be achieved by a move to the open-source software products. The fundamental problem is nobody orchestrates the holistic aspects of computing. As a result, nobody assumes responsibility for driving the world of computing deeper and deeper into the black hole of complexity. On the contrary, the complexity has been exploited in profit-making ventures.

Achieving stable, trustworthy, holistic products for general-purpose platforms is within the realm of known technology [2, 3]. In fact, the most frustrating aspect of the IBM and Wintel domination is that there have been several developments during these eras that could have resulted in significantly more stable platforms.

The key is to employ an architecture based upon well-structured function distribution between hardware and software as a means of reducing total system complexity and assuring security [4]. The world is awaiting the actor(s) to accomplish this task and move the industry toward a new era of trustworthy systems.

A move toward maturity can be measured to some extent when, among other high-level instructions, "open" file and/or "open" view instructions with arguments providing access rights and user keys are implemented via read-only microcode. This imple-

Is this era of Wintel platform dominance coming to an end? Most likely. The big question is: What will follow?

mentation would concretely exclude hacker manipulation of hundreds (even thousands) of lines of complex program code. This is one important example of providing trustworthy systems by proper function distribution.

Given the critical role of computing in our society there is an enormous market for trustworthy platforms. To be successful, platforms must be transformed from the current unnecessary complexity situation (which I have termed "busyware") to what can be called "stableware" [5]. How can this be achieved? As mentioned earlier, the most ideal situation involves a rebirth of the industry. However, there are other potential scenarios that could result in moving to trustworthy products.

A Dominant New Actor?
One possible scenario is that a new dominant actor emerges. The actor sees the holistic aspect of the hardware-software spectrum and integrates efforts to reduce complexity and market Stableware platforms. Will it be a U.S. actor? Maybe not.

There are several countries possessing the technical ability to achieve a holistic solution. This would be a perfect challenge for the European industry. However, it is not clear it can collectively get its act together; the track record in this regard is not

promising. It may happen in a single country, like Sweden or Finland. The Indian and Chinese computing industries have most prerequisites to take on this role. They are not burdened by the poor complexity track record of the U.S. computer industry; they have basic hardware technologies available, inexpensive and highly qualified software talent, as well as a cadre of ex-patriot experts. The Japanese and Korean industries could be other alternatives. The Russian computing industry has an early history of developing holistic hardware-software approaches. These countries possess the technical talent. However, all lack sufficient worldwide marketing clout.

A Dominant Customer?
An alternative scenario is that a dominant acquirer through "intelligent acquisition" will provide the impetus for change. As mentioned, during the adolescence era, government spending, particularly for military and space applications, led to industry advances. With the absolute requirement to produce trustworthy stableware platforms based upon well-defined architectures, it could be done, at least in the U.S., perhaps in Europe. Even though the current market situation is not at all like it was in the first two eras, serious computer industry suppliers would listen to a large customer.

This scenario could be politically initiated in response to a major catastrophe; for example, in transportation, power, finance, intelligence systems, or other critical areas. From a statistical point of view, the probability of a major catastrophe continues to grow in proportion to the amount of unnecessary complexity in computer-based systems.

Recently, the U.S. government announced a program for new critical system development with significant government spending being targeted for the IT industry. We may be on the way back to the government spending approach of the 1950 and 1960s. The major question is if significant advances are made through intelligent acquisition, how soon will they find their way into the critical worldwide computer and communications infrastructure?

Viewpoint

Rebirth of the Computer Industry?

By far the most beneficial scenario is a rebirth of the computer industry where the goal of producing and marketing stableware products and services in a highly competitive environment dominates. In this scenario, industry standards for system software, particularly for properly structured secure operating systems, very high-level programming languages, and databases, evolve. These standards are verifiable, and suppliers must have their products third-party certified before they are approved for the marketplace. Such certification is performed in other critical industries by testing agencies and there is no reason why it could not be done for the computer industry. As mentioned earlier, there is a history of doing this for the Cobol programming language. Certification resulted in significant competition in providing good Cobol compilers. Industry standards of the current era such as Ethernet and TCP/IP, and more recently, XML have also succeeded in stimulating useful competition.

In all three of the vital areas (operating systems, very high-level programming languages, and databases) an effort is required to define exact syntax and semantics in the form of high-level machine instruction sets that can be implemented by appropriate mixes of hardware, firmware, and software. Products that implement these standard machines are verifiable and lead to a rational basis for certification. Given such publicly available definitions and certification procedures, many existing as well as new suppliers would compete in the platform marketplace.

There have been developments in all three areas that can be used as starting points for moving toward the desired standardization. XML has become an industry database standard. For high-level programming languages, UML (although it contains unnecessary complexity) has a core that could be used as a basis for achieving a significant lift in programming abstraction level. Alternatively, there are interesting Web programming languages evolving. The C and the C++ languages are simply too low level. While Java already provides exact syntax and machine semantics, it is still too low level. Historically, languages like Simula, Modula, and parts of Ada provided useful higher-level concepts. There is some hope the Eiffel programming language can fill this high-level role.

The toughest nut to crack is the operating system. Developments must lift above the Windows-Linux debate and result in a stable verifiable open standard. High-level Web Services protocols such as Simple Object Access Protocol (SOAP) and Universal Description, Discovery, and Integration Protocol (UDDI), as well as related enterprise integration servers, may be a good starting point for providing high-level portions of the concrete semantics that is required. The world is awaiting a generic network-operating system. Who is going to provide it? Perhaps a highly modified version of Linux would be a useful starting point. Otherwise, the most impressive operating system ever provided for general-purpose computers—namely, MULTICS, developed at MIT in the late 1960s—would be an excellent candidate for inspiration [6].

Microsoft .NET

Microsoft, via .NET, is attempting to take the lead in defining Web Services core technologies. After discounting much of the hype around .NET it is evident Microsoft plans to integrate around the Internet and has an ambitious plan to do so. It has presented key technologies from .NET like Microsoft Intermediate Language (MSIL) and its new programming language C# (C sharp) for European Computer Manufacturers Association standardization. In the case of MSIL, it is in the form of an instruction set for an abstract higher-level language machine. Concepts from the 1960s and early 1970s are finally catching hold. It's about time.

It remains to be seen if the .NET technology will serve as a basis for open competition. Since the high-level .NET products can operate on non-Microsoft operating systems such as Linux and Solaris, it may be the catalyst to move from the Microsoft platform dominance. Due to the openness of Microsoft in respect to .NET, one can speculate that Microsoft strategy involves a successive retreat from being the dominant platform supplier with the plan to be a dominant supplier of high-quality network applications and content. This would be beneficial for all vested-interest parties and certainly ease the burden

upon Microsoft in respect to antitrust litigation.

Further, it remains to be seen if Bill Gates will succeed in transforming Microsoft from its tradition of continually producing more and more functionality with increasing complexity into an organization aimed at making the correct architectural decisions and utilizing the proper product life cycle processes needed to provide trustworthy systems.

IBM Autonomic Computing

As noted earlier, it seems IBM has awoken to the black hole of complexity problem it originated during the second computer industry era. As a measure of current complexity problems, Paul Horn notes that IBM has been adding about 15,000 people per year to its service organization to assist customers in dealing with complex platforms. This situation cannot continue. Horn has proposed an ambitious university research program as well as an intercompany effort based upon standards that will lead to improving the management of computer system platforms. This effort is based upon an intelligent middleware that promotes flexibility in providing IT services in a utility-like manner. I hope IBM understands this middleware should be used as a basis for defining rational function redistribution between hardware and software as noted here. Currently, the approach seems to be leaning toward adding to the software mountain via the use of AI techniques to manage existing platform complexities.

A positive indication from both Microsoft and IBM is the admission that the complexity problem leading to insecure systems is not parochial but shared by all computer industry actors. Let us hope they really mean this and they and other major actors including Sun and HP, as well as non-U.S. actors, take all appropriate actions required to correct the current situation. Of course, these actions must involve a transition strategy for moving from today's products to the standardized and certified products of the future.

Telecommunications Industry Alternative

As the world moves toward mobile Internet applications, there has arisen a plan to standardize system software functions for 3G mobile communication devices. It would be wise to agree upon concrete syntax and semantics as I've noted here. However, early efforts focus upon the use of Java and Symbian's operating system. Major telecommunication suppliers like AT&T, Nokia, Sony-Ericsson, and Motorola could potentially lead the way to a rebirth of the computer industry.

Regardless of whether Microsoft, IBM, or the telecommunications industry establishes the basis for rebirth, this scenario would stimulate worldwide competition in providing highly integrated hardware-software stableware platforms. At the same time, having standardized platforms available that do not require significant attention to expensive busyware, the market for good applications (valueware) would explode. Stableware and valueware competition would provide opportunities for existing as well as new suppliers around the globe. If this transpires, the general-purpose computer industry would be well on its way to maturity and society would be the benefactor of trustworthy products and services.

THERE ARE SEVERAL PATHS THAT CAN LEAD TO THE successive reduction of Wintel platform dominance. So, let us hope the days of the march into the "black hole of complexity" are numbered. The complexity of many of the applications for which valueware is constructed is quite significant. We do not need to continue to compound the complexity by utilizing unnecessarily complex unstable and insecure platforms of the Wintel era. Has the rebirth of the computer industry begun? Let us hope so. **C**

REFERENCES
1. Horn, P. Announcement of the autonomous computing program; www.research.ibm.com/autonomic/
2. Lawson, H.W. Salvation from system complexity. *IEEE Computer 31*, 2 (Feb. 1998).
3. Lawson, H.W. Infrastructure risk reduction. *Commun. ACM 41*, 6 (June 1998).
4. Lawson, H.W. Function distribution in computer system architectures. In *Proceedings of the Third Annual Symposium on Computer Architecture* (Jan. 1976).
5. Lawson, H.W. From busyware to stableware. *IEEE Computer 31*, 10 (Oct. 1998).
6. Organick, E.I. *The Multics System: An Examination of its Structure*. MIT Press, Cambridge, Mass., 1972.

HAROLD W. (BUD) LAWSON (bud@lawson.se) is the managing director of Lawson Konsult AB in Stockholm, Sweden.