The Impact of Software Growth on the Electronics Industry

Michiel van Genuchten
Eindhoven University of Technology/NXP Software

The amount of software in many electronic products is increasing rapidly. For example, the number of lines of source code in a mobile phone is expected to increase from 2 million today to 20 million by 2010; a car will contain 100 million lines of code (R.N. Charrette, “Why Software Fails,” IEEE Spectrum, Sept. 2005, pp. 42-49). Consequently, electronics companies no longer find it economically viable to provide all the software in their products.

As an article in The Wall Street Journal pointed out (“Shakeout Could Jolt Electronics,” 27 Jan. 2005), similar forces pressured the computer industry two decades ago:

Until the 1980s, a handful of giant manufacturers controlled the design, construction and sale of their machines, along with most of the software that ran on them. But then came along the personal computer, which relied on standardized chips and software.... In consumer electronics, parts for TV sets, digital cameras and music players are becoming so standardized and so easy to mass produce that anyone can crank out products at low cost.

This trend is evident in other electronics industries as well, including the automotive and medical electronics industries.

FROM HARDWARE TO SOFTWARE

The impact of software growth on the electronics industry was evident a decade ago (R. Gal and M. van Genuchten, “Release the Embedded Software; the Electronics Industry in Transition,” Int’l J. Technology Management, June 1996, pp. 33-44) and can be separated into five distinct stages:

Hardware. At this stage, software is not yet required to control the hardware. A few hardware-only products can still be found in today’s consumer electronics industry, such as the water cooker.

Embedded software. In products with embedded software, such as TVs and copy machines, the software is not sold separately from the hardware—that is, it does not appear on the invoice. The changes that embedded software bring within an electronics company are limited. More software engineers come on board, but the sales channels, value chain, and development strategy are typically unaffected.

Proprietary software. As the amount of software increases, embedding it is no longer cost-effective. Proprietary software appears on the product invoice but only works on the same company’s hardware. IBM was the first company to enter this stage when it began selling software separately from its hardware in 1969.

Open systems software. As software use continues to expand, individual companies cannot afford to develop all the required software by themselves. This often leads to an industry or open standard—such as Unix or Windows in the computer industry—that lets multiple vendors provide parts of the software stack. The business fundamentally changes because it must find new customers for the software and develop new sales channels. Power accordingly shifts from the electronics companies to the software companies. The computer industry first entered this stage when Compaq launched its PC in 1983, enabling startups like Microsoft to sell their software on hardware from multiple vendors.

The “Natural-Motion Software” sidebar describes a technology that was once available in hardware and as embedded software and has now become available in the form of open systems software.

Open source software. At this stage, software is only developed once and then shared with the world. This is often an attractive alternative for electronics companies that face increasing...
software development costs but are not in the software business. The open source movement began in 1991 when Linus Torvalds made his first version of Linux available.

Figure 1 compares the mobile phone and consumer electronic industries to the computer industry with respect to these five stages of software growth. Note that not all electronics products will enter the open systems or open source stage; how individual businesses evolve will depend on the actions of all stakeholders.

The computer industry took most of the 1980s to enter the open systems stage. There are indications that the mobile and consumer electronics industries are headed that way much faster. One reason is that the software industry did not exist in the 1980s, while today dominant software companies are deliberately penetrating the mobile phone and consumer electronics domains.

**LESSONS OF PHILIPS SOFTWARE**

In mid-2003, Philips Electronics formed a special division to manage sales of its software, particularly for use in multimedia products, where most of the company’s intellectual property was tied up. Philips Software focused on the mobile phone market, then entering the open systems stage, and within three years saw its software appear in 100 million devices.

To illustrate the company’s dramatic change in direction, the largest software deal in 2006 was 100 times that in 2004. Capitalizing on this success, last year Philips decided to spin off part of its software business into a new company, NXP Software (www.software.nxp.com).

One major lesson Philips learned is that a company’s emerging software business can negatively impact the sales of its much larger hardware divisions—software sales should therefore be independent of hardware sales. For example, Philips Electronics used to embed its echo cancellation software for mobile phones in its silicon; by selling it to competitors of the company’s own silicon division, Philips Software increased revenue for R&D, thereby enabling the company to maintain its industry edge.

Philips also observed that the main differences between a hardware and a software business are in the sales and marketing aspects. An electronics company knows how to make quality software products; otherwise, it wouldn’t have survived the embedded stage with millions of lines of software in its products. However, it lacks the expertise to market and sell software, which involves pricing, positioning in a new value chain, and various legal issues.

Finally, Philips eventually recognized that competition in the software industry is fierce. Staying close to the mother company’s roots and intellectual property is recommended. Moving away makes it difficult to compete with small, fast-moving software startups or companies with lower operating costs—for example, in India or China.

**MANAGEMENT OPTIONS**

As software growth impacts the electronics industry, what can managers do?

**Assess the stage of your industry.** Several trends may indicate that your hardware business is changing into a software one. First, repeated discus-

---

**Natural-Motion Software**

Video enhancement is a key technology in the TV industry. One such technology, *frame interpolation*, calculates synthetic frames between real frames to provide a higher frame rate and thus a smoother picture (G. de Haan, “Progress in Motion Estimation for Video Format Conversion,” *IEEE Trans. Consumer Electronics*, Aug. 2000, pp. 449-459). Originally developed for high-end TVs and implemented in embedded architectures, this technology is now available as open systems software and used in mobile phones and personal computers.

A problem in mobile phones is that the communication pipe does not allow the transmission of enough high-quality frames to ensure a smooth, clear picture. Natural-motion software on a handset uses only a few high-quality frames to create the required intermediate frames.

The same algorithms can be used on PC architectures to increase the frame rate from 24 fps (the standard frame rate for movies) to 60 fps (the refresh rate for computer monitors). Applying the algorithms reduces the judder experienced when playing a DVD on a PC.

---

**Figure 1. Software growth.** Within the mobile phone and consumer electronic industries, software growth today is following a pattern similar to that in the computer industry two decades ago.
sions within management about the increasing software budget can be a signal that the product is becoming too software intensive to remain in the embedded software stage. In the embedded and proprietary software stages, software is a cost issue and a burden for general management; it is not a source of revenue or profit, nor does it contribute to line managers’ bonuses.

In addition, independent software vendors might start offering components that are relevant for your products and markets. These vendors will also have other customers, very likely including your competitors. A third indication is the emergence of new and unlikely competitors, ranging from small startups focusing on a profitable niche, such as image-enhancement software for medical diagnostics or electronic program guides for TVs, to large billion-dollar companies targeting a major industry such as mobile phones. Current profits and market shares are no guarantee that your business will not be impacted—for example, IBM had peak profits and a large market share in all kinds of computers in the 1980s when software began dominating the industry.

Get software on your invoice. Software remains an engineering issue until it shows up on the product invoice, which indicates that your company is entering the open systems stage. Software now becomes another way to sell your intellectual property. Because your software business will be competing with your existing hardware business in the short term, your company should create a new, independent division. Having a dedicated sales force and assuring the independence of legal and reporting lines are critical.

Software growth has significantly impacted the computer industry and is changing many other electronics industries. It is not a question of whether intellectual property will be sold in the form of software, but which companies will do it successfully. The actions electronics companies take today will ultimately determine how large a role they will play in a world increasingly dominated by software. ■

Michiel van Genuchten is a professor in the Department of Technology Management at Eindhoven University of Technology as well as senior director of emerging software business at NXP Software. Contact him at genuchten@ieee.org.

Sponsors

ICSM is the major international conference in the field of software and systems maintenance, evolution, and management. The main goal of ICSM is to promote interaction between researchers and practitioners. The theme of this year is Moving to Service Oriented Architectures. ICSM 2007 in Paris will address these new scenarios and their major challenges of maintenance and evolution. We invite high quality submissions to all conference tracks.

Research Papers: Research papers should describe original and significant work in the research and practice of software maintenance and evolution. Case studies, empirical research, and experiments are particularly welcome. We also welcome papers that present leading edge and novel ideas in maintenance. Papers must not exceed 5000 words (10 pages IEEE style) in length, in English.

Working Sessions: We invite proposals for working sessions on any topics related to software maintenance. Working sessions are to be designed around a specific theme and be interactive and discussion oriented. Submissions should not exceed two pages in proceedings format.

Doctoral Symposium: We welcome submissions of young researchers that have delivered their Ph.D. dissertation in the last two years. Submissions should not exceed four pages in proceedings format.

Industry Track: Proposals for presentations of industrial applications are welcome. These can be state-of-the-art descriptions, experience reports and survey reports from real projects, industrial practices and models. Submissions should not exceed four pages in proceedings format.

Tool Demos: We invite proposals for tool demonstrations on any topics related to software maintenance. Demonstrations will be in an open session to allow individual interaction with the participants. Submissions should not exceed two pages in proceedings format.

Important Dates

Research Papers: April 6, 2007; 11:59pm (Apia Time)
All Other Proposals: May 4, 2007; 11:59pm (Apia Time)
Notification of Authors: June 1, 2007
Camera-Ready Copy: July 6, 2007

General Chair
Francoise Ilanans, Universite Paris 8, France

Program Co-Chairs
Gerardo Canfora, University of Sannio, Italy
Ladun Tahvildari, University of Waterloo, Canada

Working Sessions Co-Chairs
Itano Muller, University of Victoria, Canada
Massimiliano di Penta, Universita di Sannio, Italy

Doctoral Symposium Co-Chairs
Kostas Kontogiannis, University of Waterloo, Canada
Harald Wertz, Universite Paris 8, France

Industry Track Co-Chairs
Ira Baxter, Semantic Designs, USA
Mario Piattini, University of Castilla-La Mancha, Spain

Tool Demos Co-Chairs
Michele Lanza, University of Lagoa, Switzerland
Margaret-Anne Storey, University of Victoria, Canada

Publicity Co-Chairs
Jens Krinke, FernUniversitat in Hagen, Germany
Ying Zou, Queen’s University, Canada

Proceedings Chair
Andrea De Lucia, University of Sannio, Italy

Finance Chair
Dave Binkley, Loyola College in Maryland, USA
CyberChair Coordinator
Damiano Distanti, Universita di Sannio, Italy
Website Coordinator
Aurélien Duvallet, University Paris 8, France
Local Arrangements Chair
Francis Rousseaux, IRCAM, France

Sponsors

IT SYSTEMS PERSPECTIVES

http://icsm07.ai.univ-paris8.fr/