Understanding the Dynamics of Software Compatibility

Giancarlo Succi, Paolo Predonzani, and Tullio Vernazza

Understanding how vendors support varying levels of compatibility between their own and competitors’ products can help you choose software wisely.

Software compatibility is a persistent headache facing IT departments, one that routinely makes change painful. Your work’s cut out for you whether your organization switches to a different vendor’s software or simply upgrades to a new version of existing software. How often have you wrestled with converting existing files and data to formats the new software can handle?

In a perfect world, all software would be compatible and you could move files freely from any application or platform to any other. Many factors work against this utopia, however. Understanding the market dynamics that drive software producers to increase or diminish compatibility among different software packages can help you purchase wisely—and give you advance warning when a troublesome conversion looms.

NETWORK EXTERNALITIES AND THE INSTALLED BASE

Economists call the effect of users placing greater value on products already used by many others network externalities (Michael L. Katz and Carl Shapiro, “Product Introduction with Network Externalities,” J. Industrial Economics, Mar. 1992; Joseph Farrel and Garth Saloner “Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation,” The American Economic Review, June 1986). Software users value their ability to share data and knowledge with each other. Data, in this context, includes the documents exchanged through network tools. Knowledge, on the other hand, relates to the process of learning the software.

Software products, by nature complex, take years to master: Just think of the many Microsoft Word or Gnu Emacs options most people never use. Altogether, the software industry exploits strong network externalities; applications that can seamlessly accept users’ existing files and present a familiar interface and command structure will have a built-in advantage over unfamiliar applications. Quantitative studies support this conclusion, showing the relevance of network externalities on software (Neil Gandal, “Hedonic Price Indexes for Spreadsheets and an Empirical Test of the Network Externalities Hypothesis,” Rand Journal of Economics, Spring 1994; Erik Brynjolfsson and Chris F. Kemerer “Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market,” Management Science, Dec. 1996).

Products can exploit network externalities through software compatibility. For example, a given product can enhance its competitiveness by accepting its competitors’ file formats: Many users might, for example, appreciate the ability to easily import Framemaker files into Microsoft Word. Compatibility also works among a single manufacturer’s complementary products. For example, a vendor might ensure that its statistical analysis tools can seamlessly access and manipulate data residing in its spreadsheet application.
A product’s installed base consists of all the people who use that product. The larger the installed base, the greater the network externalities. To take greater advantage of network externalities, a firm must increase the installed base for its product or product line—most obviously by increasing the number of copies sold.

Another approach involves exploiting a preexisting installed user base by making the product compatible with an already popular product. The target product may belong to the same company, as happens when a firm introduces a new version of its software that is backward compatible with previous versions. Or the target product may belong to a different company, as happens when a firm introduces a product able to import files created by a competing product. In this case, the firm producing the newer product tries to capture the installed base of the existing product, encouraging a mass user migration.

The options for leveraging software compatibility expand significantly when we consider combining more than one product at a time.

**BUNDLING: 1 + 1 = 3**

The practice of combining and selling two or more products in a single package is known as bundling. Generally, a bundle’s sale price is lower than the sum of the individual products’ costs. Bundling products offers an additional marketing option that can generate greater overall sales. For example, let’s say you bundle products A and B, charging less for the two products combined than buyers would pay if they bought both A and B separately. The number of copies of A and B purchased individually, and in the A+B bundle, will likely exceed the number of copies sold if you offered only A or B for sale individually. The bundle makes buying A+B more appealing to those users who would normally buy only one of the products. For example, a user who needs A but not B because a better competitor to B already exists, may buy A+B just because of the bundle. This commercial motivation makes bundling profitable per se (John Y. Bakos and Erik Brynjolfsson, “Aggregation and Disaggregation of Information Goods: Implications for Bundling, Site Licensing and Micropayment Systems,” *Proc. Internet Publishing and Beyond: The Economics of Digital Information and Intellectual Property*, [Au: Publisher, publisher’s city and state or country?], 1997 <apparently it was in press, but the publication has been cancelled. See URL: http://mitpress.mit.edu/book-home.cfl?isbn=0262581590; The paper is available as PDF at URL: http://www.stern.nyu.edu/~bakos/adig.pdf>);

Bundling offers a further advantage, however—it taps into the installed bases of all the bundled products, even if those products are not strictly related. For example, if A is a software tool with a large and growing installed base, and B is a new product with no installed base, bundling increases B’s installed base at A’s pace, which will be higher than B’s pace alone. Moreover, if A and B are compatible with and complement each other, bundling enforces the effects of network externalities between them.

The dynamics of bundling and compatibility in general are, however, affected by a firm’s position relative to its competitors.

**EXPLOITING THE DOMINANT POSITION**

Incompatibility moves the market as forcefully as does compatibility. Compatibility attracts users; incompatibility repels them. Generally, firms strive to reap the benefits of compatibility while leaving the drawbacks of incompatibility to competitors. Unfortunately, incompatibility is a double-edged blade that can damage firms that wield it. Market leaders who occupy the dominant position, however, can exploit incompatibility more safely.

A market leader may take advantage of its products’ large installed base to cut off competitors by purposely making its products incompatible with competitors’. Although both the leader’s and its competitors’ products are incompatible with respect to each other, the competitors’ products are “more incompatible” because of their smaller installed base.

Although the dominant firm’s goal remains the same—to provide the seamless interoperability the user needs—its strategy for achieving that goal becomes market dominance. Instead of offering its users full compatibility with all products on the market, the dominant firm strives to make its products the ones that everybody uses, achieving full interoperability through monopoly.

Obviously, monopoly is not a goal most firms in established markets can realistically expect to achieve. Smart and agile firms can, however, leverage the accelerating pace of technology to create a market, impose a monopoly on it, and expand aggressively from a solid installed base.

**COMPATIBILITY FACTORS**

We do not yet understand the optimal criteria that firms use to make compatibility decisions. Known factors that influence compatibility decisions include the

- technology’s age,
- firm’s size relative to the market,
- product’s existing installed user base, and
- standards that apply to the product.
Can you please give some examples of how the interplay of these factors influence firms’ decisions to offer more or less compatibility?

Small companies are more likely to adopt widely used, well established technologies, both to minimize their risks and to be connected to a large installed base. Large or new companies may go for new technology to differentiate their profile with respect to the competition. An interesting discussion on strategies related to compatibility factors can be found in the book “Information Rules: A Strategic Guide to the Network Economy,” by Hal R. Varian and Carl Shapiro, Harvard Business School Press.

Multifaceted, software products offer many degrees of usually asymmetrical compatibility: That product A can use product B’s data and services does not automatically imply the reverse. Some existing situations indicate that a monopoly can exist. We must still assess whether, as a counterweight to the benefits of compatibility, users attach some negative value to a monopoly. Nicholas Economides’ research provides evidence that monopolies, or firms that enjoy dominant market positions, may scare and even repel potential buyers (Nicholas Economides, “The Economics of Networks,” Int’l J. Industrial Organization, Mar. 1996). External actors neither directly connected to users nor to producers, such as regulatory bodies, may also play a significant role in buyer perceptions. For example, if a government decides to accept electronic documents in a given proprietary format, it endorses such format and the associated tools: users may expect to be requested to supply documents in such format or may think that such format will never be discontinued.

Most compatibility effects rely on timing: Introducing compatible or incompatible products causes different results depending on the state of technology and the market. For example, there is a known trade-off (Joseph Farrel and Garth Saloner, “Standardization, Compatibility, and Innovation,” Rand J. Economics, Spring 1985) that makes quickly adopting new technologies a high-risk, high-profit action. The software market will always move as fast as its operational environment changes. Thus, to be even partially valid, any general theory about compatibility in the software market must stretch its assumptions regarding time to their limits.

Giancarlo Succi is an professor at the Department of Electrical and Computer Engineering of the University of Alberta. Contact him at Giancarlo.Succi@ee.ualberta.ca.

Paolo Predonzani is a doctoral student at the Dipartimento di Informatica, Sistemistica e Telematica of the Università di Genova. Contact him at predo@dist.unige.it.

Tullio Vernazza is an associate professor at the Dipartimento di Informatica, Sistemistica e Telematica of the Università di Genova. Contact him at tullio@dist.unige.it.