"A NOTE ON BACKWARD COMPATIBILITY"

by

Michael KENDE*

93/75/EPS

* Assistant Professor of Economics, at INSEAD Boulevard de Constance, 77305 Fontainebleau, Cedex, France.

Printed at INSEAD, Fontainebleau, France

A Note on Backward Compatibility

Michael Kende *

Assistant Professor of Economics

INSEAD

October, 1993

Abstract

A system innovator with a monopoly over the production of both hardware and software chooses to make a new system not backward compatible with the software of the preceding generation when profits on sales of new software outweigh the development costs.
JEL Classification Number: L12

*Michael Kende, INSEAD, Boulevard de Constance, 77305 Fontainebleau Cedex, France. I would like to thank Patrick Rey and Lars-Hendrik Röller for their input.

1 Introduction

Many consumer electronics products are actually systems consisting of a hardware component and complementary software. This division leads to compatibility issues. Matutes and Regibeau 1988) show how compatibility can benefit system producers by enabling consumers to "mix and match" between components of rival contemporaneous systems. Compatibility issues also arise between successive generations of technologies. An innovator introducing a technically superior hardware product may choose to make the product incompatible with existing software. For example, Nintendo's16-bit Super NES home video-game machine is not able to play the games of the older 8-bit NES generation. One factor distinguishing Nintendo from most system innovators is that Nintendo licenses and receives royalties on all software sold. Under certain conditions to be examined here the revenues from the sales of replacement software for a new generation of hardware outweigh the development costs.

2 Model

In this model the new hardware technology is vertically differentiated from the old, where s is the parameter indexing the quality of the new technology and $\bar{s}$ indexes the old technology, with $s > \bar{s}$. There are two types of consumers with differing tastes for technology. There are λ consumers of type θ_1 and $1 - \lambda$ consumers of type θ_2, where $\theta_2 > \theta_1$. It is assumed that the market for the old technology is fully saturated; each consumer i has one unit of hardware and $\bar{N}_i$ units of

differentiated software. The reservation utility consumer i derives from the old system is:

$$\bar{U}(\bar{N}_i) = \theta_i \bar{s} \bar{N}_i^{\beta} \tag{1}$$

where $0 < \beta < 1$ so that utility is concave in software.

If the new hardware is not backward compatible with old software, consumers who purchase a new system are assumed to have unit demands for one unit of hardware and N_i units of software. Software is assumed to be symmetric. Net surplus from buying the new system is:

$$S_i^{NBC}(N_i) = \theta_i s N_i^{\beta} - p_0 - pN_i - \bar{U}(\bar{N}_i) \tag{2}$$

where p_0 is the price of hardware and p is the price of software. It is assumed that $\bar{N}_i \geq N_i$; because the market for the old technology is saturated consumers already own at least as much software as they would buy for the new non-compatible system. Software is assumed to be symmetric. From equation (2), maximizing utility in the amount of software gives:

$$p(N_i) = \beta \theta_i s N_i^{\beta - 1}. \tag{3}$$

Given a common price of software, $\beta \theta_1 s N_1^{\beta-1} = \beta \theta_2 s N_2^{\beta-1}$. Therefore,

$$N_1 = \left(\frac{\theta_2}{\theta_1}\right)^{\rho} N_2 \tag{4}$$

where $\rho = \frac{1}{\beta - 1}$. Substituting (3) into (2) and rearranging terms shows that given unit demands for hardware, each type of consumer buys if:

$$(1 - \beta)\theta_i s N_i^{\beta} - \theta_i \bar{s} \bar{N}_i^{\beta} \geq p_0. \tag{5}$$

If the new hardware is backward compatible, consumer surplus is:

$$S_i^{BC}(\bar{N}_i) = \theta_i s \bar{N}_i^{\beta} - p_0 - \bar{U}(\bar{N}_i) \tag{6}$$

and consumer i buys if:

$$\theta_i s \bar{N}_i^{\beta} - \theta_i \bar{s} \bar{N}_i^{\beta} \geq p_0. \tag{7}$$

The innovator of the new hardware technology is assumed to have a monopoly on this technology. If the technology is not backward compatible there are sales of software and the monopolist is assumed to be able to exclude third-parties from producing software.[1] The monopolist in this case follows the standard two-part tariff pricing model (Tirole, 1988, pp. 143-148), in which given the price of software the profit-maximizing hardware price (fixed fee) equals the residual surplus of the type-θ_1 consumer. It is assumed throughout that both types of consumers are served. Therefore, from (5):

$$p_0(N_1) = (1 - \beta)\theta_1 s N_1^{\beta} - \theta_1 \bar{s} \bar{N}_1^{\beta}. \tag{8}$$

[1]This is similar to the model found in Church and Gandal (1992); however they study a vertically integrated hardware manufacturer in the context of rival contemporaneous (incompatible) hardware technologies.

Given this hardware price the type-θ_2 consumer purchases and has excess surplus which is partially spent on software, so $N_2 > N_1$. Each variety of software has a fixed cost F and marginal cost c. I assume that N_2 types of software are produced, from which type-θ_1 consumers purchase a random assortment of N_1. Therefore, total sales of each of the N_2 types of software are:

$$x = \lambda \frac{N_1}{N_2} + 1 - \lambda = \lambda \left(\frac{\theta_2}{\theta_1}\right)^{\rho} + 1 - \lambda \quad \text{using (4)} \tag{9}$$

Profits of the monopolist are:

$$\pi^{NBC} = p_0(N_1) - c_0 + N_2[(p(N_2) - c)x - F], \tag{10}$$

where the monopolist sells both hardware and all software for the system.[2]

Substituting for N_1 from (4) and maximizing π^{NBC} in N_2 gives:

$$N_2^* = \left(\frac{F + cx}{\beta\theta_2 s\left[(1-\beta)(\frac{\theta_2}{\theta_1})^{\rho} + \beta x\right]}\right)^{\rho}. \tag{11}$$

If the innovator chooses to be backward compatible with old software there is no software production for the new hardware given the assumption that the market is already saturated with the software of the preceding generation. The price of the new hardware is once again set equal

[2]In general it is not always the case that the monopolist would choose to sell all software if possible. Kende (1993) shows within a dynamic framework the conditions under which it is profit-maximizing for the monopolist to not produce any software. Allowing monopolistic software competition in the future acts as a credible commitment to low future prices for software, increasing the incentives of consumers to incur the setup cost of purchasing the main component. In the static model presented in this paper, software expenditures directly reduce the price of hardware; non-backward compatible hardware sales with third-party production of software can never be as profitable for the monopolist as hardware sales with backward compatibility.

to the surplus of the type-θ_1 consumers; from (7):

$$p_0(\bar{N}_1) = \theta_1 s \bar{N}_1^{\beta} - \theta_1 \bar{s} \bar{N}_1^{\beta}. \tag{12}$$

Profits in this case equal:

$$\pi^{BC} = p_0(\bar{N}_1) - c_0. \tag{13}$$

Consumer surplus equals:

$$CS^{NBC}(N_1, N_2) = \lambda S_1(N_1) + (1 - \lambda) S_2(N_2). \tag{14}$$

and

$$CS^{BC}(\bar{N}_1, \bar{N}_2) = \lambda S_1(\bar{N}_1) + (1 - \lambda) S_2(\bar{N}_2), \tag{15}$$

depending on the type of system. The two-part tariff scheme is always set up so that the type-θ_1 consumer is indifferent between purchasing the system or not, thus $S_1(\cdot)$ always equals zero. Therefore, regardless of whether or not the system is backward compatible consumer surplus equals $(1 - \lambda)S_2(\cdot)$.

3 Results

The system innovator chooses to be non-backward compatible when the profits outweigh those under backward compatibility. Figure 1 below shows the conditions under which non-backward

compatibility is chosen. The assumption is that consumers already have at least as much software as they would buy under non-backward compatibility. Figure 1a on the left shows the results when $\bar{N}_i = N_i$ and Figure 1b on the right when $\bar{N}_i > N_i$. Each figure shows the demand curves of a type-θ_1 and type-θ_2 consumer. Linear demand curves are used for expositional simplicity.

In Figure 1a the total area A and B represents the gross surplus for each type-θ_1 consumer of using the new hardware with the old software. This area represents the fixed price of the backward-compatible hardware gross of the reservation utility of the old system.[3] If the system is non-backward compatible, new software is sold at price p^*. The type-θ_1 consumers now spend B on software, which reduces the fixed price proportionately to area A. Regardless of the type of system the total expenditure of the type-θ_1 consumers is identical and net surplus is zero. If the system is not-backward compatible the type-θ_2 consumers each spend A on the hardware and B and C on software. Therefore, non-backward compatibility increases revenues by C from each of the $(1-\lambda)$ type-θ_2 consumers. However, in order to earn these extra revenues the monopolist must create N_2 new varieties of software at a fixed cost of F per variety. Under backward compatibility the type-θ_2 consumers each receive surplus of areas C and D; with non-backward compatibility the surplus is reduced to D due to the software expenditures.

Figure 1b represents the profit comparison when $\bar{N}_i > N_i^*$. The hardware price, p_0, under backward compatibility is areas A, B, and C. Under non-backward compatibility price p^* is charged for each variety of software, reducing the hardware price to A. The type-θ_1 consumers each spend B on software and the type-θ_2 consumers spend B, C and D on software. With non-

[3] I ignore the reservation utility (1) of the old system for simplicity because it cancels out of the profit comparisons since it is identical for both types of systems.

backward compatibility revenues increase by area D from each of the $(1-\lambda)$ type-θ_2 consumers, but decrease by area C from the expenditures of the λ type-θ_1 consumers. The net gain must be compared with the cost of software development. The surplus of the type-θ_2 consumers under non-backward compatibility is now only E compared with the surplus of areas D, E, and F under backward compatibility.

These results lead to the following propositions:

Proposition 1: For $\bar{N}_i \geq N_i^*$ the monopolist chooses non-backward compatibility when the following condition is satisfied:

$$(1-\lambda)p(N_i^*)[N_2^* - N_1^*] - [\theta_1 s\bar{N}_1^{\beta} - \theta_1 s N_1^{*\beta}] \geq N_2^* F. \tag{16}$$

Proof: The first term on the left-hand side of (16) equals area C and D in Figure 1b; additional software expenditures by the type-θ_2 consumers. The second term equals area C; note that $(1-\lambda)C$ cancels from the first term so that the net gain from non-backward compatibility is $(1-\lambda)D - \lambda C$. Non-backward compatibility is chosen if additional revenues are greater than total software development costs. Note that if $\bar{N}_i$ equals N_i the second term in (16) disappears and the first term is simply area C in Figure 1a.

From (16) any parameter increasing the extra expenditures on software makes non-backward compatibility more likely. The greater is $(1-\lambda)$, the proportion of type-θ_2 consumers, the greater are software revenues under non-backward compatibility. The greater is θ_2 the greater is the type-θ_2 consumers' demand for replacement software. As θ_2 increases areas C and D both increase; the

net effect on the left-hand side of (16) depends once again on the proportion of type-θ_1 consumers to type-θ_2's. The greater is $(1-\lambda)$ the more likely (16) holds as θ_2 increases. On the other hand, the greater is $\bar{N}_i$ the more profitable is backward compatibility due to the increase in the hardware price that consumers are willing to pay in order to consume the new hardware with the old software. Finally, the greater is the software development cost, F, the more costly is non-backward compatibility.

In terms of consumer surplus:

Proposition 2: Consumer surplus is always greater with backward compatibility.

Proof. By simplification it is straightforward to show that $CS^{BC}(\bar{N}_1, \bar{N}_2) > CS^{NBC}(N_1, N_2)$ as long as $\bar{N}_2^{\beta} > (1-\beta)N_2^{*\beta}$. This is always true given $\bar{N}_2 \geq N_2^*$ and $\beta < 1$.

Regardless of the type of system, the type-θ_1 consumers earn zero surplus. As shown above in Figure 1, as long as $\bar{N}_i \geq N_i^*$ the type-θ_2 consumers derive less surplus under non-backward compatibility due to the expenditures on software replacement.

4 Conclusion

This paper examines the conditions under which an innovator introducing a new technology which is comprised of hardware and software would not make the system backward compatible with the old generation's software. If the system is not backward compatible, consumers purchase new software, and an innovator such as Nintendo which can exclude third-party software may earn more profits than with backward compatibility. If consumers are forced to replace software in

order to enjoy the new technology, under certain conditions additional revenues derived from the sales of replacement software outweigh development costs and non-backward compatibility is chosen by the innovator, at the expense of consumer surplus.

References

CHURCH, JEFFREY AND NEIL GANDAL, 1992, "Integration, Complementary Products and Variety," *Journal of Economics and Management Strategy*, 1: 651-675.

KENDE, MICHAEL, 1993, "Profitability under an Open versus a Closed System", INSEAD mimeo.

MATUTES, CARMEN AND PIERRE REGIBEAU, 1988, "'Mix and Match': Product Compatibility without Network Externalities," *RAND Journal of Economics*, 19: 221-234.

TIROLE, JEAN, 1988, The Theory of Industrial Organization, (MIT Press: Cambridge, MA).

Derivation of Propositions; For Reference Only.

Proof of Proposition 2:

$$CS^{BC}(\bar{N}_1, \bar{N}_2) > CS^{NBC}(N_1, N_2)$$

$$\lambda S_1^{BC}(\bar{N}_1) + (1-\lambda) S_2^{BC}(\bar{N}_2) > \lambda S_1^{NBC}(N_1) + (1-\lambda) S_2^{NBC}(N_2)$$

Because the pricing is such that $S_1(\cdot) = 0$;

$$S_2^{BC}(\bar{N}_2) > S_2^{NBC}(N_2)$$

$$\theta_2 s \bar{N}_2^{\beta} - \theta_2 \bar{s} \bar{N}_2^{\beta} - p_0 > (1-\beta)\theta_2 s N_2^{*\beta} - \theta_2 \bar{s} \bar{N}_2^{\beta} - p_0$$

$$\theta_2 s \bar{N}_2^{\beta} - (\theta_1 s \bar{N}_1^{\beta} - \theta_1 \bar{s} \bar{N}_1^{\beta}) > (1-\beta)\theta_2 s N_2^{*\beta} - ((1-\beta)\theta_1 s N_1^{*\beta} - \theta_1 \bar{s} \bar{N}_1^{\beta})$$

Using (4) (and assuming old software was sold in similar proportions) and setting $\theta_1 = 1$;

$$\theta_2 s (1 - \theta_2^{\rho}) \bar{N}_2^{\beta} > (1-\beta)\theta_2 s (1 - \theta_2^{\rho}) N_2^{*\beta}$$

$$\bar{N}_2^{\beta} > (1-\beta) N_2^{*\beta}.$$