Supporting Services and the Choice of Compatibility

Chien-fu Chou* and Oz Shy** †

April, 1989
Revised, January 1990

Abstract

This paper investigates the incentives of firms to produce compatible products in a market where consumers' choice among brands depends on the availability of complementary services supporting the individual brands. We analyze a computer industry in which consumers treat computers and computer specific software as complementary products. Our main result shows that, despite the fact that a computer firm can increase the variety of its supporting software by making its machine compatible with other machines' software, a firm may end up losing part of its market share and profit by investing in compatibility. This paper also underlines the importance of distinguishing between hardware compatibility and economic compatibility.

Keywords: INSERT KEY WORDS

JEL Classification Number: INSERT NUMBERS

[0]*State University of New York at Albany, Albany, NY 12222.

[0]**State University of New York at Albany, Albany, NY 12222, and Tel-Aviv University, 69978 Tel-Aviv, Israel.

[0]†We thank two anonymous referees for most valuable comments.

1. Introduction

The paper analyzes firms' pricing behavior and their incentives to produce compatible products when the competition over their market shares is affected by the availability of complementary products. Our analysis applies to complementary products such as VCRs and videotapes, computers and software, and cameras and accessories. Our approach is different from the one taken by the network externalities literature, see for example Katz and Shapiro (1985), in the following way.[1] Here, we do *not* assume that utility of consumers increases with the number of users of the same brand. Instead, we assume that consumers have preferences for commodities and the variety of supporting services.[2] We analyze a computer industry in which consumers treat computers and computer specific software as complementary products.[3] This model applies also to other industries in which consumers do not care about the number of consumers purchasing the same (compatible) brand, but choose a brand according to prices of all brands and the amount of complementary goods supporting each brand.[4]

We develop a model for a computer duopoly industry in which the production of complementary software requires a large fixed cost of development relative to the cost of duplicating and marketing the software. The consumers' expenditure level on software and the increasing returns to scale in software production determine the actual amount of software produced for each machine. The amount of software available for

[1]Chou and Shy (1990) shows that even *without* assuming network externalities the utility of a consumer can increase with an increase in the number of consumers buying the same brand. Later, in Chou and Shy (1989b), we have shown the possibility of negative network effects in the sense that an increase in the number of users may reduce the welfare of existing users.

[2]The role of supporting services is also discussed in Swann (1987).

[3]Economides (1989a,b) and Matutes and Regibeau (1988) provide an alternative approach in which firms produce all the components of the systems and these components are compatible or incompatible with other systems. In the present framework, hardware and software are produced by different industries, where the software industry consists of many monopolistically competitive firms.

[4]For example, the Beta and VHS video recorders are supported by incompatible video tapes. Compact disk players and turntables are also supported by incompatible disks. For comprehensive discussions of compatibility issues see Farrell and Saloner (1985, 1987).

each machine and the price charged by each computer firm determine which machine is purchased by consumers. In this framework each computer firm is aware that if the systems are incompatible then cutting its price has two effects: It increases its market share and the profitability of its supporting software industry. This in turn increases the variety of its specific supporting services and hence the desirability of the system, thereby increasing the computer firm's market share. Therefore, this framework emphasizes that the existence of monopolistically competitive supporting industries increases the incentives of computer firms to cut prices.

In this paper we investigate computer industry equilibria under three hardware designs: two-way compatibility, one-way compatibility, and incompatibility. Two-way compatibility occurs when both computer firms make their machines compatible with the other's software. Incompatibility occurs when no firm invests in compatibility. One-way compatibility occurs when only one firm makes its machine compatible with the other's software. We find out that one-way compatibility may result in two types of equilibria. One in which machine specific software coexists with common software (software compatible with both machines), and the other in which only common software is produced. If in a one-way compatibility equilibrium only common software is produced, we say that the machines are *economically compatible*. Otherwise, we say that the machines are *economically one-way compatible*.

Our framework is useful to explain why in some industries firms do not invest in making their products compatible. The reason for this is that if one firm has an incentive to make its product compatible with the other's software then one-way compatibility equilibrium may result in a situation where the machines are economically compatible, thereby eliminating the incentive of the second firm to invest in compatibility. Perhaps the most striking result obtained in this framework is that paradoxically a firm which makes its computers compatible with the other machine's software may lose some of its market share and reduce its profit. This case can happen despite the fact that this firm's customers enjoy two types of software after the machine becomes compatible.

The paper is organized as follows. In section 2, we develop a basic model and analyze the computer and the software industries. In section 3 we analyze a computer industry equilibrium and demonstrate the existence of unique globally stable equilibrium price strategies of computer firms for the cases where computers are compatible or incompatible. In section 4 we analyze the incentives of firms to produce compatible products. Section 5 concludes.

2. The Model

We consider a two company computer industry. The industry produces two brands named Artichoke (brand A) and Banana (brand B). There is a continuum of potential software firms producing computer specific software packages. For each computer brand i, $i = A, B$, there is a continuum of potential software packages indexed by x_i, $x_i \in [0, \infty)$. Thus, x_i denotes a particular software package for machine i, $i = A, B$. The set of actually produced software packages for machine i is denoted by X_i, which is assumed to be Lebesgue measurable in $[0, \infty)$. Thus, the number of actually produced software packages is the Lebesgue measure of X_i denoted by $\mu_i = \mu_i(X_i)$. The quantity produced/consumed of software package x_i is denoted by $s(x_i)$. The economy consists of a continuum of consumers uniformly indexed by δ, $\delta \in [0, 1]$, where δ also represents the consumer's preference type to be discussed later in this section. Each individual is endowed with L dollars to be spent on computers and software.[5]

[5]Using a Cobb-Douglas utility function defined over an outside good and computer systems, we can extend our results to cases where consumers' expenditure on systems is not predetermined.

2.1 Technology

Each computer company i produces computers under constant cost of M_i dollars per computer, $i = A, B$. Each software firm operates under increasing returns to scale. The production of one piece of software for computer i, $x_i \in X_i$, $i = A, B$, requires a fixed cost (the fee paid to the software writer) f_i, and a unit marginal cost which can be interpreted as the cost of duplicating and marketing one unit of a software package (say, the cost of a computer diskette). We associate each piece of software with a single firm.

2.2 Consumers

A consumer derives utility from computer systems. We define a system i, $i = A, B$, as one computer of brand i *and* a collection of machine i software packages $\{s(x_i),\ x_i \in X_i\}$, where $s(x_i)$ is the consumption level of package x_i for a consumer who owns this system.[6] Since a system contains only one computer, the service of a system is proportional to the amount of software packages in the system. The service of a system i is denoted by S_i. Formally, the service of a system i for an individual is defined by

$$S_i = \left\{ \int_{X_i} [s(x_i)]^\alpha dx_i \right\}^{\frac{1}{\alpha}}, \quad \text{for } i = A, B, \ \ 1/2 < \alpha < 1 \tag{1}$$

The restriction of α to be greater than 1/2 is needed to ensure well defined reaction functions of the two firms, see equations (9), (10), and the lemma. We define the utility of an individual type δ by

$$U^\delta = \begin{cases} \delta S_A & \text{if he is an Artichoke user} \\ (1-\delta) S_B & \text{if he is a Banana user} \end{cases} . \tag{2}$$

Thus, a consumer indexed by a high δ is Artichoke computer oriented and a consumer indexed by a low δ is Banana oriented.

[6] We choose to work with perfectly divisible software packages in order to make the degree of software substitution well defined.

2.3 Market Structure

The computer hardware industry is assumed to behave as a price setting duopoly, where each firm takes the other firm's price as given.

The software industry consists of a continuum of software firms, where we impose a Chamberlinian monopolistic competition market structure implying monopolistic pricing and zero profits associated with free entry. Thus, in this market structure, the number of software products (firms) μ_i, $i = A, B$, adjusts to make the profit of each software firm equal to zero.

2.4 Consumers' selection of systems

Given the CES system service function (1), the monopolistically competitive price of a software package is given by $1/\alpha$. Observe that the consumer's consumption of software package x_i compatible with system i is the expenditure on software i (denoted by E_i) divided by the price of a package and the number of existing packages for machine i. Thus,[7]

$$s(x_i) = (\alpha E_i)/\mu_i, \quad i = A, B. \tag{3}$$

Hence, the service of a system i is proportional to the expenditure on software included in system i and is given by

$$S_i = \left\{ \int_{X_i} [s(x_i)]^\alpha dx_i \right\}^{\frac{1}{\alpha}} = (\alpha E_i)(\mu_i)^{\frac{1-\alpha}{\alpha}}, \quad i = A, B. \tag{4}$$

Denote by P_i the price of computer i, $i = A, B$. If a consumer chooses to purchase a system i then its expenditure on the software compatible with system i is given by $E_i = L - P_i$. Therefore,

$$S_i = [\alpha(L - P_i)](\mu_i)^{\frac{1-\alpha}{\alpha}}, \quad i = A, B. \tag{5}$$

[7] Here all consumers purchase the same amount of each software package. This need not be the case if we assume heterogeneous consumers with different software service functions.

We denote by $\hat{\delta}$ the consumer who is indifferent between the two systems. From (2) we have that $\hat{\delta}$ is determined by solving $\hat{\delta}S_A = (1-\hat{\delta})S_B$. Hence, by (5), given the prices charged by the two computer firms, P_A and P_B , after some manipulations we have that

$$\hat{\delta} = \frac{S_B}{S_A + S_B} = \frac{1}{1 + \left(\frac{L-P_A}{L-P_B}\right)\left(\frac{\mu_A}{\mu_B}\right)^{\frac{1-\alpha}{\alpha}}} \tag{6}$$

Thus, a consumer type $\delta < \hat{\delta}$ will purchase system B, while a consumer type $\delta > \hat{\delta}$ will purchase system A, see also figure 1. Since consumers are uniformly distributed on [0, 1], the total number of Artichoke users is given by $\delta_A \equiv 1 - \hat{\delta}$. Similarly, the total number of Banana users is given by $\delta_B \equiv \hat{\delta}$.

INSERT FIGURE 1

2.5 Software industry equilibrium

The total revenue of a (i compatible) software firm is given by $\delta_i s(x_i)(1/\alpha)$ which is the product of the total number of users (δ_i), the quantity demanded by each individual ($s(x_i)$), and the price of a package. The total cost of a software firm is $\delta_i s(x_i) + f_i$. Given the price of computer i, the total expenditure on software compatible with computer i by each user of system i is $E_i = L - P_i$. Therefore by (3), $s(x_i) = \alpha(L-P_i)/\mu_i$. Now, using the monopolistic competition free entry equilibrium condition, the number of software firms producing software for system i (μ_i) can be found by equating the profit of each firm to zero. Hence,

$$\pi(x_i) = \delta_i s(x_i)\left[\frac{1}{\alpha} - 1\right] - f_i = \frac{\delta_i(1-\alpha)(L-P_i)}{\mu_i} - f_i = 0, \quad i = A, B. \tag{7}$$

From (7), using $\delta_A = 1 - \hat{\delta}$ and $\delta_B = \hat{\delta}$, it follows that

$$\frac{\mu_A}{\mu_B} = \left(\frac{L-P_A}{L-P_B}\right)\left(\frac{f_B}{f_A}\right)\left(\frac{1-\hat{\delta}}{\hat{\delta}}\right). \tag{8}$$

2.6 The profit of computer firms

Since each consumer buys only one computer, the profit of each firm is the price of a computer minus unit cost times the firm's market share. Substituting (8) into (6), after some manipulations we have that

$$\hat{\delta} = \frac{1}{1 + \Phi \left(\frac{L - P_A}{L - P_B} \right)^{\theta}} \quad \text{where} \quad \Phi \equiv \left(\frac{f_B}{f_A} \right)^{(\theta - 1)/2} \quad \text{and} \quad \theta \equiv \frac{1}{2\alpha - 1} > 1. \tag{9}$$

Therefore, the profit of computer firms A and B is given by[8]

$$\Pi_A = (P_A - M_A)(1 - \hat{\delta}) = (P_A - M_A) \left[1 - \frac{1}{1 + \Phi \left(\frac{L - P_A}{L - P_B} \right)^{\theta}} \right] \tag{10}$$

$$\Pi_B = (P_B - M_B)\hat{\delta} = (P_B - M_B) \left[1 - \frac{1}{1 + \Phi^{-1} \left(\frac{L - P_B}{L - P_A} \right)^{\theta}} \right]$$

3. The equilibrium market shares

3.1 Incompatible systems

We now describe the computer industry equilibrium. First, each computer firm sets its price taking the other's price as given. Consumer observe computer prices (P_A and P_B) and the variety of software available for each machine (μ_A and μ_B) and choose which system to buy. At the same time each software industry adjusts according to the aggregate expenditure on each type of software. Hence, in the first stage each computer firm chooses its price taking into consideration its effects on market shares (6) and the variety of software available for each machine (7). All these effects are incorporated into the profit functions (10). Thus, in some sense computer firms lead the software industries since pricing decisions are affected by the feedback of the supporting software industry.

[8] Observe that equations (8), (9), and (10) are not defined for $P_A = P_B = L$.

A computer industry equilibrium is the pair $\{P_A,\ P_B\}$ such that P_i maximizes Π_i, $i = A, B$. By (9), the equilibrium prices determine the equilibrium market shares of the two firms. In this subsection we show how the computer market is divided between the two firms when the systems are incompatible. First, we show that the computer firms' reaction functions are well defined. Then, we show that there exists a unique computer industry equilibrium.

We need the following lemma. The proof is given in the appendix.

Lemma 1 *Let* $F(x; k, \lambda) \equiv x\left[1 - \dfrac{1}{1 + k(\lambda - x)^\theta}\right]$ *where* $k,\ \lambda > 0$ *and* $\theta > 1$. *Then,* $F(x)$ *attains a unique maximum on* $[0,\ \lambda]$.

Define $\lambda_i \equiv L - M_i$, and $k_A \equiv \Phi(L - P_B)^{-\theta}$ and $k_B \equiv \Phi^{-1}(L - P_A)^{-\theta}$. Then, $\Pi_i = F(P_i - M_i; k_i, \lambda_i)$, $i = A, B$. By the Lemma, given P_B there exists a unique $P_A \in [M_A, L]$ which maximizes Π_A. Similarly, given P_A there exists a unique P_B which maximizes Π_B. Therefore, we can define firm i's reaction function with respect to the price charged by firm j, P_j, and denote it by $P_i = R_i(P_j)$, $i, j = A, B$, and $i \neq j$. The reaction functions are implicitly defined by the first order condition $0 = \partial \Pi_i / \partial P_i$, $i = A, B$. After some manipulations, the first order conditions become

$$L - P_A + \Phi(L - P_B)^{-\theta}(L - P_A)^{\theta+1} - \theta(P_A - M_A) = 0 \quad \text{and} \tag{11}$$

$$L - P_B + \Phi^{-1}(L - P_A)^{-\theta}(L - P_B)^{\theta+1} - \theta(P_B - M_B) = 0.$$

Using (9), (11) can be written in a form relating computer prices to market shares. Thus,

$$\hat{\delta} = \theta\left(\frac{L - P_A}{P_A - M_A}\right) \quad \text{and} \quad (1 - \hat{\delta}) = \theta\left(\frac{L - P_B}{P_B - M_B}\right). \tag{12}$$

Equation (12) shows that, other things equal, a firm with a larger market share will charge a higher price. From (11), the slopes of the reaction functions are given by

$$\left.\frac{dP_B}{dP_A}\right|_{R_A} = \frac{(1+\theta)D_A}{\theta} \frac{1}{\Phi\left(\frac{L-P_A}{L-P_B}\right)^{1+\theta}} \quad \text{and} \quad \left.\frac{dP_B}{dP_A}\right|_{R_B} = \frac{\theta}{(1+\theta)D_B} \frac{1}{\Phi\left(\frac{L-P_A}{L-P_B}\right)^{1+\theta}}, \tag{13}$$

where $D_i \equiv 1 + k_i(L - P_i)^\theta$, and Φ is defined in (9). From (13), we have that the reaction functions are upward sloping implying that the price strategies are strategically complements, see Bulow et al. (1985) or Tirole (1988, Ch. 5). In addition, from (13) we have that in equilibrium $dP_B/dP_A|_{R_A} > dP_B/dP_A|_{R_B} > 0$, which implies that the system of reaction functions is globally stable. Hence, if an industry equilibrium exists, then it is unique. To establish existence observe that $R_i \to L$ and $R_i' \to \infty$ as $P_j \to L$. Also, $R_i(M_j) > M_i$. Therefore, the two reaction functions must intersect at an interior point, see figure 2.[9]

INSERT FIGURE 2

We can now state the following proposition.[10]

Proposition 1 *There exists a pair of price strategies $\{P_A^*, P_B^*\}$, $M_i < P_i^* < L$, which constitutes a unique globally stable computer industry equilibrium. Also, these price strategies uniquely determine the equilibrium market shares of the two firms.*

3.2 Two-way compatibility

We now suppose that computer firms decide to produce machines which run the same software. The next section discusses the incentives of achieving compatibility. Thus, there will be only one software industry producing a variety of software measured by μ. We denote by f the (fixed) cost of developing a new software package. Consumers are still ranked according to their relative preferences towards a particular system.[11] We can think of IBM compatible computers where some users have preferences for desktop machines while others lean towards laptop machines. Observe that under

[9] In view of the previous footnote, $P_A = P_B = L$ cannot be an equilibrium.

[10] Basic comparative statics results regarding the effects of changes in hardware and software costs are given in an earlier version of this paper, Chou and Shy (1989a).

[11] Otherwise, if consumers view all computers as identical, then this market structure is reduced to the traditional Bertrand equilibrium with marginal cost pricing.

compatibility $\mu = \mu_A = \mu_B$. Hence, in view of (6), the market share of firm B becomes

$$\hat{\delta} = \frac{1}{1 + \left(\frac{L-P_A}{L-P_B}\right)} \tag{14}$$

Notice that since both machines use the same software, the market shares are independent of the software substitution parameter α. Comparing (14) with (19), the profit functions of computer firms can be derived by substituting $\Phi = \theta = 1$ into (10). Hence, proposition 1 applies also to the compatibility case implying that a unique globally stable equilibrium exists. Thus, under compatibility the firms' reaction functions are given by

$$L - P_A + (L - P_B)^{-1}(L - P_A)^2 - (P_A - M_A) = 0 \quad \text{and} \tag{15}$$

$$L - P_B + (L - P_A)^{-1}(L - P_B)^2 - (P_B - M_B) = 0.$$

Now, if both computer firms have identical production costs ($M_A = M_B \equiv M$) then $\hat{\delta} = 0.5$, and the equilibrium prices and profits are given by

$$P_A = P_B = \frac{2L + M}{3} \quad \text{and } \Pi_A = \Pi_B = \frac{L - M}{3}. \tag{16}$$

3.3 One-way compatibility

Suppose that Artichoke machines are "more advanced" in the sense that they can run both Artichoke and Banana software, but Banana machines can only run Banana specific software. In some sense this is similar to the upward compatibility concept of Hergert (1987), where a new computer model can read the old model's software. However, in the present case the A-machines do not necessarily run faster. For example, some Apple machines can run some UNIX software. In this case, the software variety available to an A-user is given by $\mu_A + \mu_B$ where the variety available for B-users is only μ_B. Thus, for given P_A, P_B, μ_A, and μ_B, firm B's market share is given by[12]

$$\hat{\delta} = \frac{1}{1 + \left(\frac{L-P_A}{L-P_B}\right)\left(\frac{\mu_A+\mu_B}{\mu_B}\right)^{\frac{1-\alpha}{\alpha}}}. \tag{17}$$

Define by $s^i_j \equiv s^i(x_j)$, $x_j \in X_j$, the consumption level of a j-software package by an i-user, $i, j = A, B$. In view of (3), $s^A_A = s^A_B = \alpha E_A/(\mu_A + \mu_B)$ and $s^B_B = \alpha E_B/\mu_B$. In view of (7), the zero profit conditions in the A and B software industries are now given by

$$\pi_A = s^A_A(1-\alpha)(1-\hat{\delta}) - f_A = \frac{(1-\hat{\delta})(1-\alpha)E_A}{\mu_A+\mu_B} - f_A \leq 0 \quad (= 0 \quad \text{if} \quad \mu_A > 0) \quad \text{and}$$
$$\pi_B = \left[(1-\hat{\delta})s^A_B + \hat{\delta}s^B_B\right](1-\alpha) - f_B = \frac{(1-\hat{\delta})(1-\alpha)E_A}{\mu_A+\mu_B} + \frac{\hat{\delta}(1-\alpha)E_B}{\mu_B} - f_B = 0. \tag{18}$$

Notice that for $f_B \leq f_A$ (A-software is more costly to develop compared with B-software) then (18) shows that in equilibrium only B- software will be written. That is, $\mu_A = 0$. Therefore, although B-machines are not compatible with A- software, the two machines are in effect compatible since they both run all the existing software. In this case the equilibrium is equivalent to the two-way equilibrium of subsection 3.2. If this case occurs, we say that the computers are *economically compatible.* Now, for $f_B > f_A$ the A-software industry may or may not exist depending on the equilibrium computer prices. Taking into account the possibility of these two cases, the analog of (8) is given by

$$\frac{\mu_A+\mu_B}{\mu_B} = \max\left\{\left(\frac{L-P_A}{L-P_B}\right)\left(\frac{f_B-f_A}{f_A}\right)\left(\frac{1-\hat{\delta}}{\hat{\delta}}\right),\ 1\right\} \tag{19}$$

Notice that if the ratio in (19) equals 1 then $\mu_A = 0$, and the machines are economically compatible since only B-software is written. Otherwise, $\mu_A > 0$ implying that some

[12]Equation (17) does not hold when the marginal costs of producing A and B software packages are not equal since in this case we cannot simply replace μ_A by $\mu_A + \mu_B$ in (5) in order to obtain the A-user's welfare level S_A.

software packages are written specifically for machine A, and in this case we say that the machines are *economically one-way compatible*. Using (17) and (19), we can verify that firm B's market share is given by

$$\hat{\delta} = \begin{cases} \frac{1}{1+\left(\frac{L-P_A}{L-P_B}\right)} & \text{if } \left(\frac{L-P_A}{L-P_B}\right)^2 \leq \frac{f_A}{f_B-f_A} \\ \frac{1}{1+\Psi\left(\frac{L-P_A}{L-P_B}\right)^\theta} & \text{otherwise} \end{cases} \tag{20}$$

$$\text{where} \quad \Psi = \left(\frac{f_B - f_A}{f_A}\right)^{(\theta-1)/2}.$$

The condition in (20) shows that if P_A is very high then the software expenditure of A-users is so small that A-software will not be produced. Thus, the two machines are economically compatible and the firms' market shares are determined similar to the two-way compatibility, see equation (14). If P_A is relatively low, then both types of software are produced and the two machines are economically one- way compatible. The profit and the reaction functions can be found similar to (10) and (11), respectively. However, in view of (20) the reaction functions have a jump discontinuity and therefore may intersect at either the economic compatibility region, $(E_A/E_B)^2 \leq f_A/(f_B - f_A)$, or at the economic one-way compatibility region, $(E_A/E_B)^2 > f_A/(f_B - f_A)$.

In summary, one-way *hardware* compatibility may result in two types of equilibria. One in which only B-software is written ($\mu_A = 0$) and the two machines are *economically compatible*. In the second type both software industries coexist ($\mu_A > 0$) and A-machines can run A-specific software which cannot be run on B-machines. In the second type equilibrium the hardware definition and the economic definition of one-way compatibility coincide and the machines are said to be *economically one-way compatible*.

4. Incentives to achieve standardization

In this section we analyze firms' incentives to invest in compatibility. First, we illustrate these incentives for the symmetric case where the two computer firms have equal marginal costs ($M_A = M_B$) and both A and B- software packages have the same development cost ($f_A = f_B$). The last two subsections consider the general case.

4.1 Incentives to achieve compatibility under symmetry

We now formulate a game to model incentives to invest in compatibility. Each firm can choose either to stay incompatible (strategy 'NC') or to invest \$ F_i , $i = A, B$, in making its machine compatible with the other machine's software (strategy 'C'). We assume that $M_A = M_B \equiv M$, and $f_A = f_B$. The payoff matrix for this game is given in table 1.

INSERT TABLE 1

If both firms play NC then the systems are incompatible and the profit of each firm is found by solving (11) for $P_A = P_B$ and then substituting into (10). If only one firm invests in compatibility, the machines become one-way compatible. Substituting $f_A = f_B$ into (18) yields that the two machines are economically compatible, where the profits can be found using (16). When both firms play C, the machines are two-way compatible and the profits are also found from (16).

Suppose that initially the machines are incompatible. From table 1 we have that firm A has incentives to invest in compatibility if

$$\frac{(L-M)(\theta-1)}{3(2+\theta)} > F_A. \tag{21}$$

Notice that LHS(21) measures the gain from compatibility while F_A is the investment cost. Obviously, if the cost of making machine A compatible with B- software is high relative to the firm's maximal profit margin $(L - M)$, then firm A has a smaller incentive to invest in compatibility. On the other hand, if consumers highly value the variety of software (α is low and hence θ is high) then the investment is more likely to occur. This can be explained as follows. It is well known that compatibility softens competition among firms (the demand faced by each firm is less elastic), and therefore other things equal, all firms have incentives to make their products compatible, see Economides (1989a), Matutes and Regibeau (1988), and Chou and Shy (1989b). When α decreases, the variety of software becomes more important and the competition between the firms intensifies (depends more on software variety) thereby making compatibility more desirable. Thus, other things equal, the likelihood of having firm A investing in compatibility increases when α decreases. Therefore, we can state the following proposition.

Proposition 2 *A firm is more likely to make its computer compatible with the other machine's software if the investment cost (F_i) is low, or when the maximal profit margin ($L - M$) is high, or when consumers highly value the variety of software available for their machines (α is low).*

Clearly, the outcome (C,C) is not an equilibrium since when one firm invests in compatibility then the machines become economically compatible thereby eliminating all the incentives for the other firm to invest in compatibility. In other words, if firm A invests in compatibility firm B becomes a free rider. Moreover, since firm B does not pay for the investment cost, the benefits for firm B from having firm A investing in compatibility exceed that of firm A.

Observe that if the software development costs are different from one machine to the other (say, $f_B > f_A$) then if only firm A invests in compatibility the reaction functions have jump discontinuities as discussed in 3.3. If the difference in software development cost is substantial then we expect the reaction functions to intersect at the one-way compatibility region. Otherwise, the intersection occurs at the economic compatibility region. In order to avoid some very tedious calculations associated with discontinuous reaction functions, the following subsections analyze the incentives of a single firm to invest in compatibility rather than deriving the equilibria for the game. Subsection 4.2 investigates the incentives of a firm to invest in compatibility when the investment leads to economic one-way compatibility. Subsection 4.3 considers the case when investment leads to economic compatibility.

4.2 Does one-way compatibility always pay?

It is generally thought that a computer firm invests in compatibility in order to increase its market share. That is, given that a machine runs software compatible with other machines, it will be able to attract more consumers and therefore increase its market share. We now show a paradoxical result in which a computer firm investing in making its machines compatible with other machines' software loses its market share even if the investment in compatibility is costless. Consider a situation where A-software's development cost is substantially lower compared with B-software's development cost so that investment results in economic one-way compatibility. Recall that Φ and Ψ measure the cost of developing a B- software package relative to developing an A-software package under incompatibility and one-way compatibility respectively. Comparing (20) with (9), we find that if $f_B > f_A$ then $\Psi < \Phi$. Hence, A-software is more costly to develop (relative to B-software) under one-way compatibility compared with the incompatibility case. Thus, A's market share can be reduced if the firm makes its machine compatible with B-software. Therefore, we state the following proposition.

Proposition 3 *If software firms still produce A-software after firm A makes its machine compatible with B-software (the two machines are economically one-way compatibility), then firm A's market share and profit decrease when A-machines become compatible with B-software.*

Sketch of Proof. Using the reaction functions (11) we can show that $dP_A/d\Phi > 0$ and $dP_B/d\Phi < 0$. Thus, by (12) A's market share $(1-\hat{\delta})$ decreases when Φ is reduced to Ψ. By (10), Π_A decreases. *Q.E.D.*

This paradoxical result can be explained as follows. When firm A makes its machine compatible with B-software, then software writers will increase the amount of software written for B-machines since this software is also purchased by A-users. On the other hand, the variety of A-software will be reduced since A-users allocate part of their software expenditure towards B-software. The attractiveness of machine A relative to machine B depends, other things equal, on the ratio of their software varieties, μ_A/μ_B, under incompatibility, and $(\mu_A + \mu_B)/\mu_B$ when firm A makes its machine compatible with B-software. Although the variety of software available to A-users increases, the substantial increase in the variety of B-software in fact reduces the ratio of software variety available to A-users compared with B-users. That is, A-computers become relatively less attractive. Proposition 2 implies that if firm A foresees that its supporting software industry will continue producing A-software, then firm A will not design its machine to be compatible with B-software.

Observe that proposition 3 holds true even if the hardware production costs are not equal $(M_A \neq M_B)$. When $M_A > M_B$, it is possible that firm A starts out with the lower market share but it may still reduce its profit and market share by investing in compatibility.

4.3 The incentives to achieve hardware compatibility

The result of the last subsection is obtained for the case where firm A initially has a very large market share compared with firm B. Here we analyze the case where firm A starts out with either a lower market share or a marginally higher market share. Thus, if firm A invests then the machines become economically compatible and only B-software is written. Observe that this is the only compatibility case which can occur in this model.[13]

We assume now that firm B does not invest in compatibility and analyze the condition under which firm A will invest in compatibility and therefore achieve economic compatibility. Assume that $M_A = M_B = 0$. From (16) we know that if firm A invests thereby making the machines economically compatible then its profit is given by $\Pi_A = L/3 - F_A$, where F_A is the (fixed) cost of making machine A compatible with B-software. Denote by $\Pi_A^*(\Phi)$ the profit of firm A when it does not invest in compatibility and the systems are incompatible as described in (10) and (11), where Φ defined in (9) summarizes the development cost of producing Banana software relative to Artichoke software. We can now state the following proposition.

Proposition 4 *If $\Pi_A^*(\Phi) < L/3 - F_A$, then firm A will invest in compatibility.*

Proposition 4 states the formal condition which yields economic compatibility. It is clear that when the cost of making machine A compatible with machine B's software (F_A) is high then the situation where firm A invests in compatibility is less likely to occur. Intuitively, when F_A is not too high, there are two potential benefits that may induce firm A to invest in compatibility. First, if the development cost for producing A-software is higher compared with B-software development cost, then firm A has a lower market share and profit relative to firm B. In this case firm A can increase its market share and its profit by making its machine compatible with B-software.[14] Therefore, if the cost of developing A-software is high (f_A is high) then firm A will

[13] By proposition 3 we have that economic one-way compatibility (a situation where one firm invests in compatibility and being supported by both software industries) cannot occur since in the case of economic one-way compatibility the investing firm is worse off. Also, in this model two-way compatibility (a situation where both firms make their machines compatible) cannot occur. This can be explained as follows. If the firm supported by software which is more costly to develop relative to other machine's software makes its machine compatible with the other machine's software then (18) implies that its software industry will vanish thereby making the machines economically compatible. Therefore, there is no incentive for the other computer firm to invest in compatibility.

[14] Here we assume that $M_A = M_B$. However, if $M_A < M_B$ then firm A may have a larger market share. Nevertheless, in this case firm A may still want to invest in compatibility if the cost of producing (developing) A-software is relatively high.

have stronger incentives to make its machine compatible with the other machine's software. Thus a situation where firm A invest is more likely to occur. Second, as discussed above, compatibility soften competition among firms and therefore all firms have larger incentives to make their products compatible when consumers' love for variety of software increases (a decrease in α , hence an increase in θ).

5. Concluding remarks

In this paper, we propose a framework for modelling the behavior of firms producing products which are differentiated because of the incompatibility of their product specific supporting services. Each firm's pricing decision is affected by the feedback of its supporting industries. In this model the benefits from investing in compatibility depend on the software development costs, the importance of variety to consumers, and the cost of investing in compatibility. On the other hand, the profitability from investing in compatibility depends less on hardware production costs although hardware production costs do affect market shares. Previous literature showed that since compatibility softens competition among firms, firms generally choose to produce compatible products. However, it is observed that many industries produce incompatible products. Our main result shows that a computer firm may not be able to increase its market share and profitability by making its machine compatible with other machines' software even though compatibility guarantees a larger variety of software available to its users. This situation can occur even if the investing firm starts out with the lower market share.

Finally, consumers here do not necessarily benefit from system compatibility. From (5) we can see that both the variety of software and the price of computers affect consumers' welfare. Thus, even if under compatibility there is an increase in the variety of software, the reduction in price competition among firms may increase the price to a certain level which makes consumers worse off. Therefore, the welfare effect of making the systems compatible is ambiguous.

Appendix

Proof of Lemma: Observe that $F(0) = F(\lambda) = 0$ and $F'(0) = 1 > 0$. Therefore, F attains an interior maximum on $[0, \lambda]$. The FOC is given by

$$1 + k(\lambda - x)^\theta = \frac{\theta x}{\lambda - x}. \qquad \text{(A.1)}$$

Define the functions $G(x) \equiv$ RHS(A.1), and $H(x) \equiv$ LHS(A.1). Observe that $G(0) = 0$, $G'(x) > 0$, and $G(x) \to \infty$ as $x \to \lambda$. Also, $H'(x) < 0$, $H(0) = 1 + k\lambda^\theta > H(\lambda) = 1$. Therefore, (A.1) has a unique solution $x^* \in (0, \lambda)$, which must be the unique maximum of F. *Q.E.D.*

References

[1] Bulow, J., Geanakoplos, J. and Klemperer, P., "Multimarket Oligopoly: Strategic Substitutes and Complements." *Journal of Political Economy*, Vol. 93 (1985), pp. 488-511.

[2] CHOU, C. and SHY, O., "Network Effects Without Network Externalities," *International Journal of Industrial Organization*, 8 (1990) pp. 259- 270.

[3] CHOU, C. and SHY, O., "Market Share Rivalry With Compatibility But Without Network Externalities," SUNY-Albany Discussion Paper #89-2, 1989a.

[4] CHOU, C. and SHY, O., "Do consumers Always Gain When More people Buy the Same Brand," SUNY-Albany Discussion Paper #89-3, 1989b.

[5] Dixit, A. and Stiglitz, J., "Monopolistic Competition and Optimum Product Diversity." *American Economic Review*, Vol. 67 (1977), pp. 297-308.

[6] ECONOMIDES, N., "Desirability of Compatibility in the Absence of Network Externalities," *American Economic Review*, Vol. 79 (1989a), pp.1165-1181.

[7] ECONOMIDES, N., "Variable Compatibility Without Network Externalities," Stanford University, mimeo, 1989b.

[8] FARRELL J. AND SALONER G., "Standardization, Compatibility, and Innovation," *Rand Journal of Economics*, Vol. 16 (1985), pp. 70-83.

[9] FARRELL J. AND SALONER G., "The Economics of Horses, Penguins, and Lemmings," in Gable L.H. ed., *Product Standardization and Competitive Strategy*, Amsterdam: North-Holland, 1987.

[10] HERGERT M., "Technological Standards and Competition in the Microcomputer Industry," in Gable L.H., op.cit..

[11] Katz, M. and Shapiro, C., 1985, "Network Externalities, Competition and Compatibility," *American Economic Review*, Vol. 75 (1985), pp. 424-440.

[12] KLEMPERER P., "The Competitiveness of Markets With Switching Costs," *Rand Journal of Economics*, Vol. 18 (1987), pp. 138-150.

[13] Matutes, C. and Regibeau P., "Mix and Match": product compatibility without network externalities, *Rand Journal of Economics*, Vol. 19 (1988), pp. 221-234.

[14] SWANN G., "Industry Standard Microprocessors and the Strategy of Second-Source Production," in Gable L.H., op. cit..

[15] Tirole, J. *The Theory of Industrial Organization.* Cambridge MA: The MIT Press, 1988.

[16] VON WIEZSÄCKER, C.C., "The Costs of Substitution," *Econometrica* Vol. 52 (1984), pp. 1085-1116.