University of Augsburg
Prof. Dr. Hans Ulrich Buhl
Research Center
Finance & Information Management
Department of Information Systems
Engineering & Financial Management

Universität
Augsburg
University

Discussion Paper WI-81

Standardization and the Competition Between Standard Business Software and Framework Technology: Policy Implications for the Management and the Standardization Organizations

by

Bernd Reitwiesner, Stefan Volkert, Melanie Frank

July 2000

in: Sprague, R. H., ed., Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Sciences (HICSS-34), Internet and the Digital Economy Track, Wailea Maui, Hawaii, (USA), January 2001, 7. volume, IEEE Computer Society Press, Los Alamitos, Hawaii, 2001

Universität Augsburg, D-86135 Augsburg
Visitors: Universitätsstr. 12, 86159 Augsburg
Phone: +49 821 598-4801 (Fax: -4899)
www.fim-online.eu

Standardization and the Competition Between Standard Business Software and Framework Technology - Policy Implications for the Management and the Standardization Organizations

Bernd Reitwiesner, Stefan Volkert, Melanie Frank
University of Augsburg, Business School, Department of Information Systems,
Universitaetsstrasse 16, 86135 Augsburg, Germany
{Bernd.Reitwiesner|Stefan.Volkert}@Wiso.Uni-Augsburg.de, {Melanie.Frank}@Student.Uni-Augsburg.de

Abstract

From a technical point of view, frameworks appear to be a promising means to provide the industry with business management software quickly and in a cost effective manner. Nevertheless, apart from IBM's San Francisco Framework, there are not really many initiatives by the software industry to develop application frameworks to compete with established standard business software vendors. When analyzing the economics of software development with or without frameworks, there are two particularities to be considered: the network effect and the effect of compatibility decisions. Our microeconomic model incorporates these two effects and allows to derive recommendations on the strategic positioning of a software vendor in this competition. We show how the results in the model change by introducing a standard. If it is possible to establish framework technology on the software market, social welfare will increase. Therefore we re-commend the standardization organizations to support framework technology by establishing a standard.

1. Introduction

Framework-based development of information systems is one of the concepts currently being discussed to be a solution to the rapidly growing demand for flexible and extendable business applications, which have to be available in reasonable time, with reasonable costs at an adequate quality level (see [3], p. 24, [1] and [14]). The vision of this concept is to be able to adopt construction principles successfully established, e.g. in the automobile industry, for the efficient construction of software: you take a platform and put the required standardized modules together according to the customers' needs. So that finally a fully individualized car - although consisting of standard components - leaves the assembly room. In software terms, the platform is the framework and the modules are the components which fit into the framework. The aim is to be able to develop business software which is as specific as individual software, but at the same costs as a standard software-based solution. Although framework-based industrial style development of individual business applications is a great vision, there are not really many application frameworks available or could be regarded as established in the market for enterprise resource planning (ERP) software. Still, standard software based solutions, especially SAP's R/3 system, dominate the market for ERP systems. So what are the obstacles, preventing this vision of an efficient, industrial production of individualized software coming true? Naturally, there are still a lot of problems to solve as far as technology is concerned ([7] and [8] give a good survey of the problems arising in this area). And from the perspective of individual framework vendors, there is some empiric evidence that several framework projects have not fulfilled the expectation of an acceptable return on investment (see [13], p. 14). Therefore, in this paper we want to take the bird's eye view onto the competition between standard software solutions and framework based solutions for enterprise resource planning systems. By analyzing the respective market positions of these two technologies by means of a microeconomic model, we want to enter into the question, whether the development of frameworks is worth being pursued: *can the framework technology get the market share necessary to allow its profitable development?*

We focus our analysis on the important market for enterprise resource planning systems. One major determining factor in this competition is the question of a standardization of the business processes supported by ERP-software and the corresponding standardization of interfaces of separate modules working together to provide the necessary functionality. Our analysis of the competition will focus on the influence of an existing vs. a not existing standard and the degree of standardization onto the relative market positions of these two technologies.

First, we have to substantiate the notion of frameworks we want to have a look at. A definition one can easily agree to is the description of a framework as a collection of interacting classes representing a reusable design of a specific software. A framework defines the architecture of an application. By completing the basic system it is possible to end up with a customer specific application. The scheme of a framework is held as generic as possible to enable fast and flexible adaptations referring to each customers' needs. The framework user completes his framework by adding the required application components. Frameworks include control flow built upon a predetermined architecture (see [4], p. 32/33).

Generally, several types of frameworks can be distinguished. The most widespread and best known frameworks are the domain-specific business frameworks, which support the development of applications in a certain problem domain. The IBM San Francisco Framework is a good example for them (for a description of the Architecture of the San Francisco Framework see [1] and [6]). We focus the discussion in this paper onto this category of frameworks.

Apart from technical and development problems mentioned at the beginning of this introduction, there are some particularities which have a great impact onto the relative market positions of the competing technologies. They are related to the network effect, which causes the utility of a consumer of a certain product to increase with the number of other users of the same product. Network effects play an important role in software markets and may result from different causes. For standard software, a potential new user of a standard resource planning system will trust the processes implemented in the software the more, the more other companies are already using the same product of a certain vendor and thus having made it being the backbone of their business. In addition to such more psychological effects, network effects may result from the need of interoperability, e.g. if two companies want to cooperate in a business network to form a virtual organization and thus being dependent on the ability to support inter-organizational business processes with the help of information systems being able to interoperate. For framework based solutions, the number of components available fitting into a specific framework will increase with the number of users of the same framework.

If two different types of standard software, two different types of frameworks, or a framework and a standard software are able to interoperate or if one of them is able to call the functionality of the other, we call them compatible (an exact definition of the notion of compatibility we want to use for the purposes of this paper will be given in chapter 2; an overview of different types of compatibility can be found e.g. in [12] pp. 253 and [16]). One product can profit from the network effects of another product, if it is partially or fully compatible. Standardization allows the different products available in a domain to be equally compatible to each other. Thus, compatibility or standardization decisions have a great impact onto the competition in software markets.

We want to analyze the competition between framework technology and standard software on the basis of a market model which allows us to incorporate the impact of network effects and compatibility decisions onto the market positions of these two competing technologies.

2. The competition between the technologies

By means of customization through parametrization standard software vendors promise to offer solutions fitting the individual business needs of the user. But regardless of its customization facilities, standard software will not allow for solutions fitting exactly to the customers' requirements and being as individual as individually developed software (see [15], p. 30). Often, customization means the adoption of the users' business processes to the reference processes supported by the standard software, but not vice versa, which of course induces substantial additional costs of restructuring the organization in part. For many emerging application domains, like supply chain management or customer relationship management, standard software is not able to provide solutions at best practice levels on time (see [10]).

On the contrary, framework based solutions allow for business applications being as individual as individually developed software by means of adoption and extension of available or insertion of additional components. One should take into account that, for example a typical application developed using IBM's San Francisco Framework consists to a degree of 40% of the framework and its related components, which are completed by an individual user interface, country and industry specifics, business rules and individual components (see [11], p. 116). So, contrary to standardized software, framework based applications can be adopted to individual needs on code level. Therefore it is obvious that a framework based application can be considered to be exactly matching the users needs. Even better, standard components can be used for standard application domains like financial accounting and individual or individualized components can be composed into the framework for the most mission critical parts of the information system, where a differentiation from competitors is desired. E.g. when looking at a bank, this could be the components dealing with risk management.

Although framework-based solutions can be regarded as optimally fitting the users needs, as opposed to standard software which is mainly designed to support the reference processes and variations coverable through customizing, standard software based solutions are not necessarily cheaper to develop than framework based solutions. This is due to the fact that in enterprise wide implementation projects of standard software the mere customization is only one of the necessary project activities. Independently from the chosen technology, in every ERP-implementation project the phases business modeling, requirements engineering, analysis and test are required. Furthermore, the costs for the redesign of the established business processes to be compatible with the reference processes of the standard solution must not be left out of consideration. Therefore, we consider the implementation of a framework based solution not to be more expensive than a standard software based solution to be a very reasonable rating of the respective cost situations (see [1], p. 58).

To be able to analyze the competition between the established technology of standard software and the (still) emerging framework technology, we have to abstract from the technical details and problems linked to framework technology and reduce our analysis to the factors determining the market position of these two technologies in the market for enterprise resource planning systems (ERP systems):

- the degree to which a standard solution meets the business requirements of a user;
- the number of users already having installed a software of a certain type and the impact of the size of this installed base on the utility of the users, usually denoted as *network effect*;
- compatibility and / or standardization decisions.

We capture these factors in a market model which is based on Hotelling's (see [5]) approach to model product characteristics as locations in a linear product space (a survey of this category of microeconomic modeling give [12] and [16]).

2.1. The Model and related assumptions

We assume the competition to occur between two suppliers *S* and *F*, both offering a complete enterprise resource planning system. *S* provides its solution as a standard software, *F* is providing solutions based on a framework and the related necessary components. We describe an oligopolistic competition, the number of participants can be reduced to 2 without loss of generality. The competition between *S* and *F* is characterized in detail by the following assumptions (the assumptions and the structure of the model follow a model presented in[9], pp. 295; we modify this model by introducing asymmetric product characteristics and by taking marginal and fixed production costs into the considerations):

(A1) Software users are considered to have requirements which can be characterized as dots on a horizontal line with the length 1. We assume the users to be uniformly distributed on this interval, so that each dot represents a single user. The position of an individual user is denoted by $h \in]0,1]$.

(A2) The total demand for software is fixed, it is completely inelastic. The suppliers compete for the size of their share of the total demand. Every consumer is supposed to buy one (marginal) unit of software. The total demand of all users is normalized to 1.

(A3) The software solutions are (horizontally) differentiated in the following way: the framework solution provided by *F* is supposed to meet exactly the requirements of any user along the horizontal line. The solution provided by *S* using the standard software approach is supposed to meet the requirements of users whose requirements are characterized by position a_S in the linear product space, i.e. the standard software is located at one point (denoted by a_s) in product space. All users with requirements different from a_S have to face "distance costs" (see figure 1), which are a linear function of the difference between the (position of the) users requirements and the (position of the) properties of the software solution.

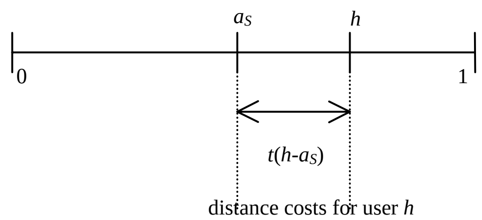

Figure 1: Distance costs in Hotelling's linear product space

The distance costs $d_S(h)$ and $d_F(h)$ can be computed as:

$$d_S(h) = t \cdot |h - a_S| \qquad \text{for the users of } S \quad (1)$$

$$d_F(h) = 0 \quad \forall h \in]0,1] \qquad \text{for the users of } F \quad (2)$$

The distance costs quantify the monetary equivalent loss of a user with requirements *h* when having to use a software solution with properties a_S. They could be interpreted as the discount a prospective customer would claim because the software does not meet perfectly his needs. *t* represents the cost unit rate per distance: the higher *t*, the more important it is for the user to have a software which exactly meets his needs and the higher is the monetary equivalent loss per unit of distance in product space.

(A4) *S* is supposed to choose the position $a_s = 0.5$ in the middle of the product space, since he maximizes the size of the symmetric interval of users on the left and the right hand side possibly choosing his solution.

(A5) Both suppliers are assumed to produce with identical marginal costs and the same costs per unit of software *c* respectively. Although we assume both suppliers already to be established in the market, there are identical fixed costs *FC* for both suppliers every period to adjust the software to current developments. These fixed costs include the costs for a possible changing of the degree of compatibility s_S and s_F respectively.

(A6) We assume duopolistic competition, i.e. the market shares of both competitors are greater than zero.

(A7) Both suppliers are supposed to maximize their profits, which can be computed as

$$\pi_l = (p_l - c) \cdot x_l - FC \quad l \in \{S, F\}. \tag{3}$$

They maximize their profits only with regard to the current period. Each supplier is fully informed about his own cost structure, the cost structure of the competitor and about the preferences and the willingness to pay of the users.

(A8) The users choose the solution with the greater consumer surplus CS_l (with $l \in \{S, F\}$). The consumer surplus consists of the basic willingness to pay *Z* less the effective price of the software solution p_l, minus the distance costs $d_l(h)$ plus the utility derived from the network effect. Thus, the consumer surplus of a user with address *h* when purchasing a solution from *F* or *S* can be formulated as:

$$CS_F = Z - p_F - t \cdot 0 + en_F \tag{4}$$

$$CS_S = Z - p_S - t \cdot | h - a_S | + en_S \tag{5}$$

$e > 0$ is a measure for the general valuation of the network effects of the users and is assumed to be equal for standard software and framework based solutions. n_l denotes the network size:

$$n_S = x_S^i + x_S^{ex} + s_S(x_F^i + x_F^{ex}) \tag{6}$$

$$n_F = x_F^i + x_F^{ex} + s_F(x_S^i + x_S^{ex}) \tag{7}$$

The network size n_l of a technology is determined by three factors:

- the size of the installed base x_l^i (i.e. the number of installations resulting from the last periods);
- the expected size of the market share in the current period x_l^{ex};
- the participation in the installed base and the expected market share of the competing technology $s_l(x_m^i + x_m^{ex})$ $(with\ m, l \in \{S, F\}\ and\ l \neq m)$.

The *degree of compatibility* $s_l \in [0,1]$ determines to which degree technology *l* participates in the number of installations of technology *m* sold in the previous and the current period. If e.g. *F* decides to be fully compatible to the technology of *S* by choosing $s_F = 1$, then his users will not only profit from the network effect of technology *F*, but will fully profit from the network effect of technology *S* as well.

(A9) The users are supposed to have complete information about the prices and the degrees of compatibility of both technologies. In particular with regard to the expected market share x_l^{ex} $(l \in \{F, S]\})$, we assume that the users have rationale expectations, i.e. the users' expectations about the market share of technology *l* exactly match the result of the competition. Formally this can be described as:

$$x_l^{ex} = x_l$$

(A10) The competition between *F* and *S* takes place in two stages: in the first stage (*compatibility competition*) the competitors decide on the degree of compatibility s_F and s_S they will choose when designing their products. In the second stage (*price competition*) the degrees of compatibility and by that the network advantages/disadvantages of the technologies are fixed and the competitors try to price their products in a way that maximizes their profits. The decisions on both stages of the competition have to be made simultaneously, i.e. competitor *F* doesn't know how *S* will decide and vice versa. The outcome of the decisions of the first stage determines the strategic position of the competitors in the price competition. The technologies can be incompatible $(s_l = 0)$, partially compatible $(s_l \in]0,1[)$ or fully compatible $(s_l = 1)$.

Beware that compatibility is not a symmetric relation. Let us, for instance have a look at the competition between the Compact Disc (CD) and the new Digital Versatile Disc (DVD), there we have asymmetric compatibility: one can't play DVD's on a CD-player, whereas it is possible to use CD's on a DVD-player. Looking at software, we have asymmetric compatibility if software vendor *A* knows the interface specifications of the software product of software vendor *B* and can make function calls to the software of *B* whereas *A* doesn't publish its interface specifications thereby not giving *B* any possibility to be able to produce compatible to *A*. For the purpose of this paper we ***define the degree of compatibility*** as follows:

- We treat the compatibility between two technologies, i.e. the compatibility between the standard software solution as a whole and the framework (including all related necessary components) as a whole. We do not look at the compatibility between the framework and its single components. We assume full compatibility between the framework and its single components as well as between the components itself.
- Partial compatibility is assumed to have the following meaning: a degree of compatibility e.g. 0.6 of software A with software B means that 60 % of the functionality of software B can be used / called up by software A.

Using these assumptions, we can deduct the demand x_F for solutions based on *F* as well as the demand x_S for solutions based on *S* in the current period. Each user chooses the solution providing the greater consumer surplus (see (A8)). Thus, a user chooses the framework if the following inequality holds:

$$CS_F \geq CS_S \Leftrightarrow |h - a_S| \geq \frac{1}{t}[(p_F - p_S) + (en_S - en_F)] \quad (8)$$

We see that users prefer the framework based solution, if their *distance* from the properties of the standard software based solution is greater than a possible ***price advantage*** (p_F - p_S) or a possible ***network advantage*** (en_S - en_F) of the standard software based solution. For matters of clarity we introduce the following simplifying assumption:

(A11) The indifferent consumer prefers the framework based solution.

If we resolve the absolute value $|h - a_S|$ with the help of a fall differentiation $(h \geq a_S \; vs. h < a_S)$ and take into account (A4) we get the demand for solutions from *F* and *S*:

$$x_F = 1 - \frac{2}{t}[(p_F - p_S) + e(n_S - n_F)] \quad (9)$$

$$x_S = 0 + \frac{2}{t}[(p_F - p_S) + e(n_S - n_F)] \quad (10)$$

Taking into consideration the assumption (A9) of rationale expectations of the consumers about the technology decisions in this period, we can rewrite the demand functions (9) and (10) to:

$$x_F = \underbrace{1}_{a} - 2\lambda[\underbrace{(p_F - p_S)}_{b} + \underbrace{e(\Delta n^i + s_S - 1)}_{c}] \quad (11)$$

$$x_S = \underbrace{0}_{a} + 2\lambda[\underbrace{(p_F - p_S)}_{b} + \underbrace{e(\Delta n^i + s_S - 1)}_{c}] \quad (12)$$

with:

$$\lambda = \frac{1}{t - 2e(2 - s_S - s_F)} \text{ intensity of competition} \quad (13)$$

$$\Delta n^i = (x_S^i - x_F^i) + (s_S x_F^i - s_F x_S^i)$$

network advantage for *S* (14)

The network advantage consists of an advantage resulting from the installed base $(x_S^i - x_F^i)$ and a compatibility advantage $(s_S x_F^i - s_F x_S^i)$. With regard to the intensity of competition we make the following assumption:

(A12) We assume the intensity of competition λ to be positive.[1]

As the demand functions are fundamental for our further investigations, let us have a look at the single terms and their economic interpretation. Furthermore, we will be able to draw some first conclusions on how the parameters will work. Interpretation of (11) and (12) will be easier if we split them up into three terms:

(a) The natural market share:

As it was assumed in (A3), software produced with the help of frameworks can be considered as tailored directly to the customer's requirements, i.e. customers are not confronted with distance costs. Therefore, a customer being asked to decide between framework based software and standard software will, when neglecting any other factors we come to later on, always choose framework based software. In our formulae this aspect is represented by the resulting natural market share of 1 for frameworks and zero for standard software.

(b) The price advantage:

For the explanation let's assume standard software is offered at a smaller price than framework based software. The greater the price difference, the more customers will decide for the standard software based solution and thereby increase the basic market share of the standard software based solution.

---

[1] By this assumption we exclude the need to take into account very specific properties of demand functions which are usually assumed for Giffen-Goods. See [9], p. 311.

(c) The network advantage:

This term incorporates the network advantage. For the term to be comprehensible, let us assume that standard software is already established on the software market and can profit from its large number of installations in former periods whereas framework based software has just entered the market. The greater the difference between the number of installations of standard software and framework based software, the greater the market share of standard software will be. Beware that F can reduce his network disadvantage by choosing a high degree of compatibility s_F, since the degrees of compatibility s_S and s_F have an impact onto the term Δn^i via the term $(s_S x_F^i - s_F x_S^i)$. While Δn^i represents the network advantage resulting from former periods, the remaining term (s_S-1) represents the impact of the sales differences in the current period onto the network advantage. Therefore, the network advantage will work for S, if the advantage from former periods is larger than the total demand of the current period (which is assumed to be 1, see (A2)) less the degree of compatibility of S: $\Delta n^i > 1 - s_S$.

Of course, (b) and (c) may also work negatively for the standard software vendor. The effects of (b) and (c) will be factored by the intensity of competition λ. This is a measure, to what extent price and network advantages influence the market share.

Now, having explained the general structure of the market share formulae, we are able to present some interesting first results:

(S1) The greater the sum of the two degrees of compatibility the smaller the intensity of competition λ.

(S2) A unilateral increase of the degree of compatibility does not have a unique impact on demand.[2]

(S3) *Fully bilateral compatibility minimizes the intensity of competition and reduces any network advantages or disadvantages to zero.*

Although this finding, at first glance, may seem weird, it can be explained easily: Fully compatible products make product features and price the only criteria for the product selection. Therefore product differences become more important thereby reducing intensity of competition. The main question raised in the introduction was the question for the market position of frameworks in the competition with standard software. Our analysis of the demand for the respective technologies yields a clear natural preference for the framework technology and shows that the standard software can get market shares greater than zero only if it can offer either price or network advantages. Our brief discussion of the market situation in chapter 1 shows, that we can assume the standard software technology to have a significant network advantage. Whether S can use the network advantage to prevent F from entering the market will be discussed in chapter 4. F can reduce the network disadvantage by choosing to produce at least partially compatible to S (i.e. $s_F > 0$). Since a market share greater than zero is not enough for a technology to be successful, we analyze the respective profit of F and S in the resulting market equilibrium in the next chapter 2.2.

2.2. Results of the price competition with given compatibility

Let us assume for the moment, that both vendors accept their degree of compatibility as given. The strategic variable left is the price of their product. As we have assumed duopolistic competition (see (A6)), it is necessary to take into account the reaction of one’s competitor when determining one's own price. The profit maximizing Nash-equilibrium price for F and S can be found at

$$p_F^B = \frac{1}{3}[3c + \frac{1}{\lambda} - e(\Delta n^i + s_S - 1)] \quad (15)$$

$$p_S^B = \frac{1}{3}[3c + \frac{1}{2\lambda} + e(\Delta n^i + s_S - 1)] \quad (16)$$

The corresponding market shares are

$$x_F^B = \frac{2}{3} - \frac{2}{3}\lambda e(\Delta n^i + s_S - 1) \quad (17)$$

for the framework vendor, and

$$x_S^B = \frac{1}{3} + \frac{2}{3}\lambda e(\Delta n^i + s_S - 1) \quad (18)$$

for the supplier of standard software. At the bottom line, the following profits can be earned:

$$\pi_F^B = \frac{4}{18\lambda} - \frac{8}{18}e(\Delta n^i + s_S - 1) + \frac{4}{18}\lambda e^2(\Delta n^i + s_S - 1)^2 - FC \quad (19)$$

$$\pi_S^B = \frac{1}{18\lambda} + \frac{4}{18}e(\Delta n^i + s_S - 1) + \frac{4}{18}\lambda e^2(\Delta n^i + s_S - 1)^2 - FC \quad (20)$$

In chapter 2.1. we have already had a closer look at the demand functions x_F and x_S. In the market equilibrium, the natural demand for framework based technology takes the value 2/3 and the natural demand for the standard software based solution takes the value 1/3. As far as a varia-

[2] This result corresponds to a result of the model we used as a basis for our analysis, see [9], p. 311.

tion of the degrees of compatibility is concerned the statements (S1), (S2) and (S3) do still hold.

When analyzing the individual profit functions, there are two relevant questions:

(a) Is the profit greater than zero? Or turns the profit maximum out to be a loss minimum?

(b) Which competitor makes more profit?

Although question (a) is difficult to answer from analyzing π_F^B (19) and π_S^B (17), we can prove both profits to be greater than zero by analyzing (15) in combination with (17) and (16) in combination with (18) respectively: we can show that prices turn out to be greater than the marginal costs *c*, as long as the market shares are greater than zero. The market shares are always greater than zero because of assumption (A6). Thus both competitors can gain a contribution margin greater than zero and therefore gain profits greater than zero as long as the fixed costs *FC* are sufficiently small.

The answer to question (b) depends very much on the set of parameters. The condition for π_F^B to be greater than π_S^B can be written as:

$$\pi_F^B > \pi_S^B \Leftrightarrow \frac{1}{6\lambda} - \frac{2}{3}e(\Delta n^i + s_S - 1) > 0 \qquad (21)$$

Again, we can state that the profit of the framework vendor will surmount the profit of the standard software vendor in case of a small intensity of competition and small network advantages of the standard software over frameworks (which e.g. can be reached by a large s_F). Hence, we can point out that for typical scenarios, where the number of software installations of *S* succeeds those of *F* to a great deal, it is necessary for a success of framework based software to be sufficiently compatible.

Thus, at the end of this chapter we are able to answer one of the questions raised in our introduction: basically, if frameworks are able to get a market share larger than zero at all, then *F* will make real profits. We were able to point out that a large degree of compatibility of *F* favors this development. But, as we discuss at the beginning of the next chapter, it may be difficult for *F* to be able to reach a high degree of compatibility with software from *S*. In the following chapter we discuss how standardization can promote the market position of frameworks.

3. Introducing a standard into the competition

In the last chapter, we assumed each of the competitors decides on his own whether his software should be compatible with other products and to what extent. But we have to bear in mind that, in case of software, a profit maximizing degree of compatibility usually cannot be reached without agreement of the vendor of the product one wants to be compatible to.

As interoperability, usually reached by compatibility, is so important, standards play a crucial role. From our software-perspective we consider a standard to be a common set of functionalities which can be incorporated within a piece of software, two or more parties have agreed on. Interoperability between software components which adopt the standard is guaranteed. The standard is adopted by a software vendor to the extent his own functionality is compatible with the common set. Software standards can be set either by a governmental non-profit-organization, or, more often by a manufacturers' consortium. We should not neglect that because of their importance in the software market, some single software vendors are able to set *de-facto*-standards. For the remainder of chapter 3 it is not necessary how the standard has been set, we come back to that question in chapter 5.

In the following we discuss the effects of a common standard within our economic model. When replacing s_F and s_S by *s* within formulae (15) to (21) the findings are applicable in this context as well, so that we do not need to repeat the analysis and can pay attention to the question of an optimal degree of standardization.

We assume both competitors to be willing to agree to a standard. Nevertheless, they might have divergent preferences as far as the degree of standardization is concerned. Thus we analyze what profit-maximizing degree of standardization each of the vendors would strive to individually in such a standardization game. Primarily, we want to know, whether in the case of a substantial network advantage of the standard software based solution, *S* will vote for low and F will vote for a high degree of standardization *s*.

We enter into that question by analyzing for which size of installed base of *S* profits of *S* and *F* increase or decrease with an increasing degree of standardization. I.e. we look for the range of the network advantage of S, where $\partial\pi_s^B / \partial s$ and $\partial\pi_F^B / \partial s$ is greater or smaller than zero. Interpreting these differentiations we get the following results:

(a) *Profit analysis of S with variable s, depending on* x_S^i*:* For a large network advantage of the standard software vendor, he will counteract ambitions to increase the degree of standardization.

(b) *Profit analysis of F with variable s, depending on* x_S^i*:* We have an area in which an increasing degree of standardization increases the profit of *F*, that is because he is able to make use of the network effect caused by *S*'s installed base.

(c) *Profit analysis for s = 1:* The profit of *F* will always be larger than the profit of *S* if it comes to total standardization. As a result, one can imagine that *S* will try to avoid that a mandatory standard will be enforced.

4. Can S prevent the market entry of the framework supplier F?

Now we will enter into the question whether there is a chance that the standard software vendor can prevent the mere market entry of the framework technology vendor. This will take place, if the market share x_F^B of F in the Nash-equilibrium (see equation (17)) is smaller than zero. To analyze that formally, we have to adjust some assumptions to this new situation of one competitor not being already established in the market:

(A13) Since we discuss a possible market entry of *F*, we can be sure that the size of the installed basis equals zero: $x_F^i = 0$.

(A14) *S* is assumed already to be established. Thus *S*, when deciding about his policy, has to take into account only the regular periodical adoption costs *FC*, which occur every period. Whereas *F* will have to take into account the full amount of development costs for the framework and the related components. We denote them by FC^{entry} and hence assume them to be greater than *FC*: $FC^{entry} > FC$.

Additionally, (A6) is not assumed to hold for the purposes of this chapter.

Without any specific assumptions with regard to the degrees of compatibility, we can basically state that the prevention of the market entry of *F* is possible, if the installed base of *S's* standard software x_S^i will be sufficiently large. This result is not at all trivial, especially if one recalls the fact that the "natural demand" for standard software solutions is zero (see equation (12)) since there are no distance costs for users of the framework technology.

We formally show that for the case of standardization. If we assume standardization, the market share of the framework vendor will be forced below zero if the following inequality holds:

$$x_F^B(s) < 0 \Leftrightarrow \frac{1}{\lambda} < e(1-s)(x_S^i - 1) \qquad (22)$$

$$x_F^B(s) < 0 \Leftrightarrow x_S^i > \frac{\frac{t}{e} - 3 + 3s}{1-s} \qquad (23)$$

We can compute the limit degree of compatibility, up to which market entry prevention is possible:

$$x_F^B(s) < 0 \Leftrightarrow s < 1 - \frac{t}{e(3 + x_S^i)} \qquad (24)$$

We see that the greater x_S^i, the greater is the possible degree of standardization *s* up to which market entry prevention is possible. Nevertheless, if standardization allows to enforce full compatibility ($s = 1$), inequality (24) will not hold for any x_S^i. Setting a standard at a high degree of standardization can thus help to keep the market open for emerging technologies. But should the market be kept open for framework technology? We will enter into that question in the next section.

5. Do frameworks increase the social welfare?

In this chapter we want to investigate the impact of the market entry of *F* onto the social welfare. The social welfare in the Bertrand-Nash Market equilibrium W^B can be computed as the sum of the consumer surplus *CS* of all users (see (A8)) and the sum of the profits of both suppliers.

$$W^B = \sum_{c=0}^{1} CS_c^B + \sum_{l=S}^{F} \pi_l^B \qquad (25)$$

Due to assumption (A2), the market is always covered and therefore the absolute prices do not have any effect on the *total* demand. Since the product $p_l^B \cdot x_l^B \; (l \in \{S, F\})$ reduces the consumer surplus with the same amount it increases the profit of the suppliers, absolute prices do not matter for total welfare W^B (see [17], p. 30). Thence we can compute social Welfare W^B as the sum of the cumulated basic willingness to pay of all customers less the cumulated distance costs plus the cumulated network effects less the cumulated marginal and fixed production costs.

We analyze that formally in the next section, thereby assuming a standard to be established.

5.1. Social welfare with an existing standard

If a standard could have been established social welfare can be computed as:

$$W^B = Z - t \cdot \left(0{,}5 - \frac{x_F^B}{2}\right)^2 + e(1+s)(x_S^i + x_F^i + 1) - c - 2FC \qquad (26)$$

We can derive the following statements from (26) regarding social welfare:

(S4) With regard to the network effect, one (marginal) unit of framework produces the same amount of welfare as one (marginal) unit of standard software.

(S5) In the case of an established standard social welfare is increasing with an increasing market share of the framework technology in the current period.

5.2. Social welfare if the market entry is blocked

When formally analyzing that problem, we regard assumptions (A13) and (A14) from chapter 4 to hold as well as (A6) not to hold. We assume a standard being established. If the market entry is blocked, the monopoly of *S* leads to social welfare amounting to W_M:

$$W_M = Z - 0{,}25 \cdot t + e(1)(x_S^i + 1) - c - FC \tag{27}$$

Since the market is always covered, absolute prices do not matter for welfare, even in the case of a monopoly. As there is only one technology, compatibility can’t enlarge the network effect. If the market entry is not blocked, social welfare will amount to W_D^B:

$$\begin{aligned} W_D^B = Z - t\left(0{,}5 - \frac{x_F^B}{2}\right)^2 + e(1+s)(x_S^i + 1) \\ - c - FC - FC^{entry} \end{aligned} \tag{28}$$

If F can enter the market, distance costs are reduced. Since we have two technologies, the network effect increases through compatibility, if the degree of standardization is greater than zero. The additional fixed costs FC^{entry} reduce social welfare. We can analyze the combination of these effects by having a look at the following inequality:

$$\begin{aligned} & W_D^B > W_M \Leftrightarrow \\ & FC^{entry} < \left(0{,}25 - \left(0{,}5 - \frac{x_F^B}{2}\right)^2\right) \cdot t \\ & \quad + es(x_S^i + 1) \end{aligned} \tag{29}$$

$$\left(0{,}5 - \frac{x_F^B}{2}\right)^2 < 0{,}25 \Leftrightarrow x_F^B > 0 \tag{30}$$

The formal analysis by means of (29) and (30) allows us to draw the following conclusions about the impact of the market entry of *F* on social welfare in the case a standard has been established:

(S6) If the fixed costs for the development of the framework technology FC^{entry} are sufficiently low (i.e. smaller than the gain in consumer surplus through the reduction of the distance costs and the additional network effect), the market entry of F and thereby the development of a new paradigm of software technology increases social welfare.

(S7) The greater the distance costs per unit *t*, the greater the fixed development costs for framework technology can be. If there is a degree of standardization greater than zero, the fixed development costs might be ceteris paribus the greater, the more the users value the network effects (expressed by *e*).

6. Implications

In the last 3 chapters we could prove that vendors of framework based software can coexist with standard software vendors. Furthermore, we were able to show that from a welfare point of view, the competition between *F* and *S* is desirable. When we consider the most likely scenario in which the framework vendor is not yet established, we can clearly state the most important condition for coexistence of both technologies in the market, thereby we deliberately neglect other less relevant effects:

Frameworks need to participate in the network effect resulting from a large number of installations of the established technology.

Now, we have to answer the question how this can be reached. Firstly, a high degree of compatibility of *F* can ensure this. But the decision of being compatible cannot be made by the framework vendor himself. Legal and technical aspects allow the established competitor to interfere. This issue may become an "explosive" political topic, as a rationally acting competitor will avoid giving up his advantages and this may in the end even lead to a market entry prevention for a prospective competitor. As we have shown in chapter 3, the adoption of a common standard by all competitors allows the potential users of the emerging technology to profit from the network effect of the established technology. But again, the incumbent won't be willing to give up his advantage. When it comes to market failures and a social welfare optimum is missed the government is asked to intervene. This could e.g. happen by promoting the emergence of public standards in national or international standardization bodies like ANSI, ISO, ITU and so on. Although legal enforcement of these standards is not acceptable, public authorities can promote the acceptance of those standards by claiming them in public contracts. But, when observing the markets of fast changing software technologies, one has to admit that public standards play a minor role only. In most cases, relevant standards are set either by the market leader or by industrial consortia. Therefore the possibilities for governmental market intervention are rather limited.

On the other hand, we have to discuss the policy implications for the standard software industry in case framework technology is able, despite of the mentioned obstacles, to gain first pieces of the market share and thereby starting to threaten the network advantages of the established standard software. As we have seen in chapter 2, the great advantage of framework based software compared to standard software technology is the exact fit to the customers requirements. Standard software vendors are beginning to copy this property by transforming their monolithic architecture towards "framework" oriented architectures, thereby allowing users to partially integrate individual components. A prominent example for this development is the SAP Business Framework (e.g. see [11], p. 113).

7. Limitations and Outlook

The model analysis applied here was designed quite simple to ensure mathematical tractability and easy interpretation. This obviously implies a number of limitations. In the following three of the most obvious limitations will briefly be discussed as well as ways to overcome them.

- In (A5) we assumed the marginal costs as well as the fixed costs for *S* and *F* to be identical. We presented arguments for this assumption to be reasonable, nevertheless different costs can be included in our model.
- We have used a very simple notion of compatibility. This is sufficient for our aim to discuss the relationship between the standard software and the framework as a whole. Economic literature provides more complex models which would allow to include a triple relationship between providers of pure frameworks, components and standard software (see [16] and [12], pp. 253).
- Our model does not incorporate time. Observing software markets, one has to agree that time-to-market plays an important role. An extension of our model could incorporate the preferences of users for time-to-market as an element of the effective price in the same way we included the distance costs.

Briefly summing up, we can conclude by giving the answer to our introductory question: "Can the framework technology get the market share necessary to allow its profitable development?" Our analysis allows to give a positive answer to that question and furthermore shows that the support of frameworks via the introduction of an accepted standard will increase social welfare.

8. References

[1] Anleiter, G. and Fent, A.: "Interoperabilität und Frameworks". *OBJEKTspektrum* 5/2000, pp. 56 - 61.

[2] Bohrer, K.A.: Architecture of the San Francisco Frameworks. *IBM Systems Journal* 37 (1998).

[3] Ferstl, O., Sinz, E.J., Hammel, C., Schlitt, M. and Wolf, S.: Bausteine für komponentenbasierte Anwendungssysteme. Höhere Mathematik und Datenverarbeitung (HMD) 197/1997, pp. 24 - 46.

[4] Gamma, E., Helm, R., Johnson, R., Vlissides, J. and Booch, G.: *Design Patterns – Elements of Reusable Object-oriented Software*, Addison-Wesley, 1995.

[5] Hotelling, H.: Stability in Competition. *Economic Journal*, 39/1929, pp. 41-57.

[6] IBM (Ed.): IBM San Francisco, http://www-4.ibm.com/software/ad/sanfrancisco/, downloaded 9/28/2000

[7] Ivanov, E.: Entwicklung und Anwendung von Frameworks – Probleme und Lösungsansätze, 1996, pp. 3-8. (http://www.theoinf.tu-ilmenau.de/~ivanov/scripts/studie.ps, downloaded 12/2/1999)

[8] Philippow, I, Ivanov, E.and Preißel, R.: A Method for the Development and Application of Frameworks. *The Third Conference on Integrated Design & Process Technology* (IDPD, incorporated ESPA, IEEE), Berlin, July 6-9 1998, IDPT-Vol.4, pp. 938-946.

[9] Pfähler, W. and Wiese, H.: Unternehmensstrategien im Wettbewerb, Eine spieltheoretische Analyse, Berlin, Springer Verlag, 1998.

[10] Reitwiesner, B. and Will, A.: Best practice oder common practice: Eine Frage der Wirtschaftlichkeit. *Wirtschaftsinformatik* 39 (1997) 6, pp. 640 - 641.

[11] Scheer, A.W.: *ARIS - Vom Geschäftsprozeß zum Anwendungssystem*. Berlin, Springer Verlag, 1998.

[12] Shy, O.: *Industrial Organization, Theory and Applications*, Cambridge (Mass.), MIT Press, 1995.

[13] Stiel, H.: Sticht Desktop-Management den Framework-Ansatz aus? – Klein anfangen, statt groß scheitern. *Computerwoche* 19/2000, pp. 14-16.

[14] Turowski, K.: Establishing Standards for business Components. K. Jakobs (Ed.) *Information Technology Standards and Standardisation: A Global Perspective*. Hershey, 2000, pp.131-151.

[15] Vaske, H.: Individualsoftware trotzt den “Modewellen” in der IT. *Computerwoche* 8/2000, pp.30 - 31.

[16] Wiese, H.: Compatibility, Business Strategy and Market Structure - a Selective Survey. In: EURAS Yearbook of Standardization, Volume 1, München, Accedo Verlagsgesellschaft, 1997, pp. 283 - 308.

[17] Woeckener, B.: Network Effects, Compatibility Decisions, and Monopolization. *Zeitschrift für Wirtschafts- u. Sozialwissenschaften* (ZWS) 119 (1999) 1, pp. 23 - 44.