DIPLOMARBEIT

Titel der Diplomarbeit

Openness and Competition in Information Technology

Verfasser

Markus Wagner BSc.

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften
(Mag.rer.soc.oec.)

Wien, am 04.08.2011

Studienkennzahl lt. Studienblatt:	A140
Studienrichtung lt. Studienblatt:	Diplomstudium Volkswirtschaft
Betreuer:	Prof. Dr. Gerhard Clemenz

Acknowledgement

First, I want to thank Prof. Clemenz for mentoring and supervising this thesis, including the discussions about the scope and direction of this work and the guidance in the modeling part. I further have to thank Prof. Müller and the student group at the diploma seminar for the interesting discussion and some helpful comments after my presentation.

I want to thank my parents Josef and Herta Wagner for their consistent financial support as well as their encouragement and guidance during my studies. The same way I want to thank my grandparents Georg and Ella Reitmair for supporting me financially and for showing interest in my work. A thank you also goes to my brother Thomas Wagner for helpful suggestions about the layout and structure and for proof-reading certain parts of the thesis.

Last but not least I want to thank my girlfriend Sigrid Fasching for always having an open ear for my problems during the work on my theses and for her consistent support and encouragement over that time.

Inhaltsverzeichnis

1. Introduction

My interest in information technology can be traced back many years when even at a very young age I was interested in electronics, computers and programming. As a graduate from a technical high school, a BSc in Electrical Engineering and current master student in Computer Technology at Vienna University of Technology and also from some work experience I gained lots of insight into information technology, probably more than most other economists were able to.

But also for pure economists information technology became very interesting in recent years: Information technology usually provides companies huge economies of scale. Information technology facilitates what is called network economies such that consumers gain welfare by the number of other consumers buying the same good, meaning that there are lock-in effects, learning effects, "platform problems" and much more. Some companies became blue prints for the economics of competition (policy) and innovation. And: There are a number of extremely successful companies, making huge profits, some already for decades but some just for very few years and many of the most successful ones are still growing.

The topic I want to do my research on is what I call "openness", a term that will be defined in more detail below. In my definition openness is a general term for topics in information technology for which a company or other organization allows access in some way to it's developments. In this work the goal is to scrutinize the properties of openness and how this relates to competition in the information technology industry.

The major research questions therefore are the following:

1. How can openness be defined in a general way and how do the different topics fit into this approach? Are there commonalities between the different topics?

2. How do successful companies use openness in their strategies and how is this

related to their success?

3. How can openness be modeled reasonably such that the features of the industry are reflected? What are the major issues and features of openness in economic terms?

The thesis is organized as follows: After the introduction there is a section in which I want to discuss important terms related to the topic and put them into a historical perspective. I will discuss information technology, it's economic features and where it came from, I will show why software is important and define openness and the topics that are related to openness as a general term. Then, in section 3 I review related work by other authors that deal with parts of my research questions or are in other ways related to my analysis. In section 4 I do an in-depth analysis into the market for information technology by going into the details of the history, strategies, products and legal issues of three very successful but however very unique companies (Microsoft, Google and Apple) and work out how they deal with openness in their strategies and history. An economic analysis follows in section 5: I develop a 4-stage model that combines network economics with a vertical product differentiation approach in a leader-follower market and discuss the properties of the model. I further show that several topics of openness can be discussed using the model by changing certain details and solve the model for the issues at hand. In section 6 the research questions will be revisited again to summerize the results and discuss possible extensions to the analysis.

2. Definitions and Historical Perspective

2.1. Information Technology

Information technology is a term that is very often used today and seems to be a public word for some time now. John (2001, p.2) states that the term is known at least since 1958: Back then the term was needed "to describe the changes in business management that would accompany the widespread adoption of the computer“. This is of course not a real definition for the term but just a description of how the term evolved. The discussion in this introduction by John is not about the term information technology but rather a discussion if today we live in an information age or not. He cites several authors that convincingly argue that information has been very important in the history of humanity and this is not tied to the modern age. Therefore, one could also go back to the 17th century as a start of the information age, maybe even back to the ancient Greeks.

Searching for a straight definition for the term information technology gives lots of different results. The most convenient I found was "the branch of engineering that deals with the use of computers and telecommunications to retrieve and store and transmit information“ (WordNet 3.0, n.d.). The definition is quite modern in a sense because computers are explicitely mentioned. This also shows that historically it was in fact computer technology that boosted information technology. Haigh (2001) shows how the connection between information and the computer worked: Instead of just using computers as machines to process data, the idea in the 1950s and 1960s was to have a managerial information system: Having a computer as an information processing tool for business people and managers. Although at that time it was not possible to build a so called MIS (totally integrated management information system), the visions led to the understanding of a computer as the main part of an information system.

The computer then not only became a vital part in dealing with information: The progress in computer technology and manufacturing was also the reason for the boom in information technology in the last decades. Starting from basic inventions like the transistor in 1947 or the integrated circuit in 1959, the progress in computer technology is linked to a technology prediction from 1965 nowadays called Moore's law: Gordon Moore, co-founder of Intel predicted that the number of transistors in an integrated circuit would double roughly every two years without higher costs (Moore, 1965)(Intel, n.d.). In the last 50 years at least this proposition held true and this allowed an exponential growth in processing speed, memory capacity, sensor capability while chip size became smaller and smaller. Those advancements led to a stream of better and more powerful products that even became cheaper over time due to economies of scale: In these industries we find large one-time investments (R&D, production sites) but very low marginal costs: Building a plant in 2009 costed about 4.2 billion dollars (Globalfoundries, 2009) while the production costs of a single modern processor is very low (there is unfortunately no good data for it).

2.2. Software

One concept that has to be understood is software with it's distinction to hardware (the actual physical device). Software can be defined as "written programs or procedures or rules and associated documentation pertaining to the operation of a computer system and that are stored in read/write memory“ (WordNet 3.0, n.d.). This is maybe not a very clear definition but the most important part is in it: It is about the actual operation of a computer system. Today, there are many different types of software: Operating systems, drivers, firmware, databases and of course application software. The main feature of software is that a developer can provide functionality more easily than in hardware due to the use of programming languages, APIs (Application Programming Interface) and compilers. One can also argue that software provides a higher abstraction level to the computer system than hardware. Cortada (2002) provides a starting point to the research on history of software and software engineering. He argues that the basics of software design and development have been laid in the 1960s and are still important today: "between 50 and 75 percent of the largest companies in any US industry used computers by 1969" (Cortada, 2002, p.73). The following rise of the importance of software is closely connected to the history of Microsoft and other companies in the 1980s which will be discussed in later sections.

2.3. Openness

I do not know of any generally accepted definition so I will define it in my own terms and explain it's relation to topics in information technology. My definition however is influenced by an article by Pontin (2009) although I want to be more precise in the definition and the relationships of the different aspects of openness.

Open Source is the one topic that maybe most people who are aware of the concept of openness might think of first (Open Source Initiative, n.d.). Here, the source code of a program is not only freely available but usually also allowed to be copied, changed and redistributed in a new product depending on the actual license. There are many different licenses available today, some that force a redistribution in the same license while others even allow the use of the source in commercial products. Some enforce the shipment of the source code with the product while others don't. Important licenses are the GPL (GNU General Public License), Apache License and the BSD (Berkeley Software Distribution) License among others. Sometimes those licenses are compatible so that a code in one license can be used in a code of another license and sometimes they are not. Given the more general approach of this thesis, the actual differences in the details will not matter in this work. Today there are several very successful products that are developed as open source, for example the Linux operating systems, the Apache webserver, several web browsers, databases and many, many more software products for smaller tasks. Lerner and Tirole (2002) argue that the history of free software started in 1983 with the development of GNU by Richard Stallman and others due to the fact that AT&T started to enforce it's property rights on UNIX. UNIX then was the most important branch of operating systems and was developed in the 1960s and 1970s in a cooperative way by academic institutions and other research facilities. Although they do not explicitely mention it, the coming of all the commercial operating systems in this time may also have played a role (see later for details when I discuss the role and history of Microsoft).

The second important aspect of openness is compatibility. A compatible system is a system that can work together with other systems or can work on or be part of different other systems. In economic terms this can be often interpreted as substitutability: A compatible system is often an alternative to an already existing system. Sometimes compatibility is seen from the perspective of an incumbent who might want to block a competitor to develop a compatible system for example by using patents. Compatibility is an issue that in terms of information technology can be divided into at least three different subissues:

1. Standards are formalizations of codes, methods, procedures, units - basically ways of doing engineering. A standard can be open in two ways: It can be openly documented which means that the documentation is available to the public or it can be openly implemented, meaning that everybody can use the standard (Välimäki, 2010). There are many institutions that collect and certify standards like the ISO (International Organization for Standardization) but standards can also be implicit for example due to a monopoly which enforces it's use. If a standard is not open it is either held secret or the implementation is protected by patents. An open standard can facilitate compatibility as it provides guidelines how to develop compatible systems.

2. Portability means that a software can work on several different operating systems or that an operating systems can work on several hardware platforms or that web sites can be shown on different web browsers and so on. A more general and partly economic definition is given by Mooney (2008): "An application is portable across a class of environments to the degree that the effort required to transport and adapt it to a new environment in the class is

less than the effort of redevelopment.“. In the last 30 years there was a lot of convergence in operating system platforms as well as in hardware platforms, meaning that it should become easier and less costly to provide portability. While in the beginning of the 1980s there were many operating systems with not much market share (see discussion in the section about Microsoft), today there are basically just three: Microsoft's Windows, Apples MacOS and Linux. Other Unix-type operating systems like Solaris, Novell Netware, HP UX and IBM's OS/2 and z/OS just play a role in special purpose and large server systems. In hardware, the personal computer market is dominated by the IBM platform and Apples system although the latter one is not so much different from the first (see the section about Apple for discussion). For smartphones or other low energy and mobile devices the ARM platform is very dominant. Other platforms like IBM's PowerPC and Cell, Suns SPARC or MIPS also just play a role in special purpose computers, high performance computing, large server systems and gaming consoles. In the last 15 years many innovations were made to make portability easier to achieve: For example Java, Flash or browser applications allow the development of portable software that works on any operating system. Hardware virtualization like for example by VMware, Parallels or Xen (open source) in principle allow the use of any operating system on any hardware platform and also the "transferability“ of certain operating system installations between different computers. Microsoft builds virtualization into it's operating systems such that applications that were developed for older versions of Windows can still be run on newer versions.

3. Interfaces are parts of a system that allow connections to other systems, hardware or software[1]. The interesting types of interfaces for this thesis are software interfaces which can be interfaces of an operating system that an application software uses or it can be an interface that a software provides for another software. Openness now means that like for standards the inter-

[1] The way a program or a computer as a whole presents itself to the user is also an interface, fittingly called user interface. This however has completely different features than the interface that connects software.

faces are documented and that there are no patents preventing the use of an interface. Many times interfaces are implicitely or explicitely standardized but it still makes sense to mention it separately from standards. Operating systems interfaces allow the use of services that an operating system provides to be used by application software, for example network transfer, file handling, memory management, window management or the use of input devices. Those are usually open via an API (Application Programming Interface) because otherwise writing software on an operating system would be very difficult and inefficient. Interfaces of an application software are usually used for communication and data transfer between applications, for example a database has an interface to obtain data, or communication software has an interface to which another communication software can connect to or a program reads data in a certain format from an input file which may have been written by another program[2].

A third aspect that I believe has to be included is licensing: A firm is open in terms of licensing when it grants competitors the right to use it's technology. This does not mean that the license has to be for free but that a choice is available for other companies. One example might be an operating system that can be licensed to other computer vendors to use the system. Another one where one company allows another company to use part of it's chip design[3]. This of course deals with issues like patents but also has to do with closed standards: A company with a superior technology and an implicitely defined closed standard could either open up by opening the standard or by licensing the standard (knowledge and/or patents) to a competitor.

And then there are also more subtopics that I want to mention quickly:

- Open Data means that a company that deals and collects information allows

[2] There can be of course a debate whether input files are a part of the software and therefore no interface to the outside world or if they are something from the outside that an application has to be able to deal with via an input interface. In this thesis it makes more sense to assume the second.

[3] This is the business plan for ARM processors.

others (users, competitors) to manage that information.

- Open Access is a way to publish information (usually online) that everyone can read, process and use.

- Reusability means that old products or parts of products can be used in the development of new products. This is clearly the case in open source development where anyone can just use code from another product as long as the license criteria are met. However, the issue is not related to open source in general as code reusability is also possible for closed source software.

Openness is therefore a very broad term that unites several subtopics that can be interesting for economists. To summerize all in a formal definition is quite hard and probably not even necessary but it is clear that openness means that an external access to a certain development is allowed by the holder of rights to that development. This can be access to code (open source, reusability), access to interfaces, standards or documentation (compatibility), access to use, customize or resell (licensing) or access from the consumer's point of view (open data, open access).

3. Related Work

Surveying economic literature that is related to my research questions I got the impression that not very much has been done in this field and the literature is scattered along different topics that are all related to mine but are clearly not seen in the same generality. To summerize and discuss those works I group them according to the relation to my thesis.

3.1. Open Source

There are two important papers by Lerner and Tirole (2002, 2005) that deal with the economics of open source and the background and reasons for the success of open source development in recent years: The diffusion of open source products, the significant capital investments by companies in open source and the new organizational structure of those projects. The interesting question that is asked is the following: "Why should thousands of top-notch programmers contribute freely to the provision of a public good?" (Lerner and Tirole, 2002, p.198).

The authors discuss several possibilities of those incentives, from altruism, to the search for external or personal benefits like future job opportunities (signaling incentives), better future access to the capital market, ego gratification or peer recognition. Benefits for a firm having employees work on open-source projects include higher motivation and improvements that can also be used in the primary (closed-source) projects although there is of course also a cost involved that some work time is lost for other projects. They also argue that companies can earn off open source by selling complementary products or even release it's products as open source especially if the product is not very profitable as closed source anyway.

From their studies of certain projects and the open source community they find that those incentives really play a role in open source development: Many open source projects were started due to insufficiency of closed source software and the inability to change this insufficiency, giving a direct return to the development that is greater than it's possible commercial usefulness. Also signaling incentives play a role which can be seen by the fact that giving credit to the author(s) is very important in the community and is enforced by several open source licenses. There is also a lot of evidence that contributors for certain open source projects were hired by commercial companies for their expertise and also open source success helped many new companies to get access to the capital market.

Another topic is the question about the relative quality of open source. There are arguments that open source has higher quality for consumers because they are able to adapt and even improve the product for their needs. Another hypothesis is that the open source process might lead to better quality because there is more peer review and even more honest review in the open source scene than in closed source developments. One argument that closed source development might lead to better quality is the assumption that in open source development documentation and user interfaces do not play a great role while those aspects are necessary for closed source commercial products.

Another issue the papers deal with is about social welfare of open source developments. They cite several papers and conclude that there is not a clear effect in one direction. The most interesting arguments are: Open source developments might bring down prices for closed source products which might lead to higher welfare but competing systems (for example operating systems) might lead to lower value of systems when less applications are available on one system. There might also be a negative innovation effect when open source software brings down profits for closed software and therefore lowers investments for innovations in that area. Open source might also have a positive impact on human capital.

Also an interesting contribution is the relationship between software patents and open source. The authors argue that patents are especially problematic for open source development as those can hardly pay royalty fees to patent holders and also do not have a defensive portfolio of patents. They also find initiatives in the open source world to insure projects against patent risks and clarifications in the open source licenses that prevent the shipment of open source products that are restricted by patents.

3.2. Network Effects and Compatibility Decisions

There is a strong literature on Network Economics and the most notable textbook seems to be the one by Shy (2004). Most related to my thesis are the first two chapters, dealing with the hardware and the software industry. The models that are developed in this book about network effects all deal with the question of compatibility of products which is one of the subtopics of openness. In the software chapter Shy develops a combined hardware-software model where consumers value hardware systems and software variety. As the offered software variety depends on the number of consumer who buy the hardware system this constitutes a network model and a compatibility analysis in a duopoly market follows. However, I am not very satisfied with the Untercut-Proof-Equilibrium approach and therefore did not base my analysis on that type of model.

There is also a paper by Church and Gandal (1992) that studies network effects of complementary products like hardware/software, operating systems/software or video game hardware/software and standardization. They argue that a hardware platform cannot be sold when there is no complementary software for that hardware and that software is not interoperable so that it can only be used on the platform it is developed for. Furthermore, there are two different hardware platforms, consumers who value variety and an exogenous number of software firms that have to decide for which platform to develop software (they cannot develop for both).

The authors show that there are two effects that point into different directions: The network effect makes a platform more valuable for a software firm when there are already many firms providing software for that platform because this leads to higher hardware sales and a bigger market for it's software products. On the other hand, the competitive effect makes a platform less attractive when there are more software firms on one platform because the firms compete against each other and (holding hardware sales constant) the profit goes down the more firms are on that platform.

Depending on the parameters of the model two equilibria are possible: One where just one platform survives, all the firms provide software for that platform and there is de facto standardization in hardware and another one where both platforms survive and software is provided for both platforms. A welfare analysis shows that the latter one might be the equilibrium even for parameters where it would be optimal to have just one platform. The model is very interesting because it combines hardware and software markets and gives a clue how those interact.

A different idea is presented in the paper by Arthur (1989) that shows that in a model with technology adoption and increasing returns there is a possibility of adoption of an inferior (long-run) technology if it gains an early lead. Historic events and lock-in are important in this kind of setting. The paper studies a timing model where agents choose a technology at specific points in time. The choice depends on preferences but for increasing returns also on the number of agents who already adopted the respective technology. If a sufficient number of agents adopted a technology then no agent that comes later chooses another technology even if it preferred that technology when there were no network effects[4]. The equilibrium is a state where all market share goes to one technology although it is not clear which technology will dominate.

Baake and Boom (2001) analyze a 4-stage game combining a vertical product differentiation and network economics model where consumers value a product by both it's inherent quality and it's network size. Here, firms first choose the product quality they want to offer, then decide on whether to install an adapter to make the

[4] It is interesting that the author actually never calls it "network effect" but "increasing returns in adoption."

product compatible to the other product, and after a price competition consumers decide which product to buy (if any). The authors show that in equilibrium, there is always the decision for an adapter and this is also welfare enhancing in this model. However, the model is quite complex as for certain parameter combinations multiple equilibria can occur in the consumer stage while there are many different solutions for equilibria in the pricing stage. This model is symmetric in the sense that both firms enter the market at the same time, although in equilibrium the firms do not set the same quality levels.

4. Market Analysis

In this section I want to do a comprehensive study on the three most successful and recognizable companies in information technology. I study the history of those companies, their products and their strategies with respect to openness in all it's facets: Microsoft, Apple and Google. Of course there are many more successful companies in information technology and all of them have a certain strategy on openness. IBM, Oracle or Intel might come to mind immediately and I do not reject that it might have been interesting to study those as well. However, it seems more useful to study less companies in more detail than the other way round and furthermore those three companies already show almost all possible strategies with respect to openness.

The first company I want to study is Microsoft, one of the most successful and profitable companies in the world. There are of course many books about Microsoft and it's history but I will primarely stick to a study by Campbell-Kelly (2001), Microsoft's own history timetable (Microsoft, 2004), the database of The History of Computing Project (n.d.) and Microsoft's product web page and documentation.

The second firm in this analysis is Apple, a firm known today for it's great profitability, it's design and product quality. Although the beginning of the company seems to be similar to Microsoft's in some aspects, the strategies, product decision, firm culture and business success were not. I primarily used the book by Linzmayer (2004), an in-depth article by Goodell (1996), interesting facts and essays at The Apple Museum (Mesa, n.d.) and information found at www.apple-history.com (Sanford, n.d.).

The third company is relatively young compared to the other two and came up when the others already were big companies and were going through ups and downs as well as different environments. Google is a child of the Internet age and became big when the Internet became more and more important. My main sources for the part about Google are it's own quite detailed corporate timeline (Google, n.d.a) and a book by Vise (2008).

Of course, the product documentations of several products of those companies as well as my own knowledge and experience play a role, too. Certainly, this study cannot be complete but it is a summary about what I think are the most important aspects of those companies in relation to my research questions.

4.1. Microsoft

4.1.1. History and Rise

Microsoft was founded in 1975 by Bill Gates and Paul Allen, it's first customer being MITS which back then produced the Altair 8800, the first personal computer in history, and Microsoft wrote some programming language (BASIC) for it. Selling it's product also to other vendors Microsoft was first a programming language company before making the decision to also go into operating systems and later application software. The company became Microsoft Incorporation by 1981 with the success story also beginning in that year, with IBM's presentation of it's personal computer and Microsoft's decision to focus primarely on the development of an operating system (and other products) for that platform. When IBM brought it's PC to the market, it consisted of an Intel processor (8086) and licensed several different operating systems that could be purchased together with the PC[5]. Although being just one of several operating systems offered by IBM for it's PC MS-DOS (Microsoft Disk Operating System) was the one to come out the most successful. Campbell-Kelly (2001) argues that it was not the best system but it was the one that was available already when IBM introduced the PC and also

[5] It is interesting that IBM which then was by far the most powerful company in the computer industry did not develop and sell it's own operating system as this type of bundling was common back then (Fisher, 2000). One reason might have been that IBM was tried by antitrust authorities in the United States due to it's bundling behavior but it also might have been due to a problem with the development schedule of the PC not allowing the development of an "IBM operating system for the IBM PC“.

the one with the best price. The awesome growth of Microsoft in the 1980s with revenues growth of 36% or more every year was clearly due to it's success with IBM type PCs of which about 90% sold were equipped with a version of MS-DOS and over 80% of sold PCs were of the IBM type by the end of the decade. Microsoft's operating systems made about a half of the yearly revenues of the company during that time.

The question is of course why no other company was able to challenge Microsoft's position in such a profitable business, especially as the earlier versions of MS-DOS were not that sophisticated (even the second version in 1983 had just about 20,000 lines of code). And then, even people at Microsoft might not have believed that MS-DOS was able to stay successful as they also bought a UNIX license from AT&T and wanted to develop a UNIX-style operating system called XENIX to become the future standard. Furthermore, one can argue that MS-DOS wasn't even the best operating systems for much of the time as in the beginning it lacked network capabilities, concurrent operation and a graphical user interface. Campbell-Kelly shows that there was a competition by up to 20 vendors offering operating systems during that time and still, nobody was able to beat out Microsoft. There are many reasons one can attribute to this success:

- The economics of increasing returns where the largest firm is also the most profitable. Microsoft had huge profits which were invested in making the product better and get into other markets as well (see below).

- The first mover advantage on the IBM platform.

- Lock-in effects where users were used to the system and faced switching costs.

- Network effects: Most applications are written for the most used operating system to have a greater potential market for the product. This in turn strengthens the dominant platform because some customers might not be able to use a needed application on another platform.

The problem that the standard in operating systems (MS-DOS) did not have the ability for concurrent operation of several programs and also no graphical interface was starting to be solved just a few years after the IBM platform was introduced. The solution was a windowing system which today of course is the standard for PCs: Several applications run in different windows concurrently and one can use a mouse to work with those windows. The idea came from Xerox already in the 1970s but the Xerox Star which was already introduced in 1981 was a commercial failure due to it's huge price ($40,000). The same is true for the Apple product Lisa which was introduced in 1983 but it also was too expensive for commercial success ($16,995). The Apple Macintosh from 1984 was a lot more successful due to it's more modest price ($2,500) and was able to get about 5 to 10% of market share in personal computers in the 1980s (see the section about Apple). The approach for the IBM platform was to not get rid of MS-DOS and develop a new system but to develop a graphical system on top of MS-DOS, as a layer between the operating system and the applications. Even some competitors in that time used MS-DOS as basis for their own operating system. This was of course also the approach that Microsoft took when announcing the development of a product called Windows in 1983. Although the computer world expected the windowing systems to become standard by 1984, all the approaches back then were commercial failures. Campbell-Kelly attributes this to the fact that the Intel processors were not fast enough at that time which made the windowing systems hard to use and customers sticked with the textual interfaces.

For Microsoft the problem was not that big though as it still had it's revenues from MS-DOS which were not threatened and it still worked on improving the Windows system in the background. In 1985 it also started a joint venture with IBM to develop OS/2 which was intended to eventually succeed MS-DOS as a standard operating system on the IBM platform. Although this also became a commercial flop when the product was finished OS/2 should become important for Microsoft later. The reasons for the flop were the higher price, the relatively modest improvements over the then-current MS-DOS version and the incompatibility with application software written for MS-DOS. Microsoft dropped out of the arrangement with IBM later which improved and sold the product further although with very modest success. Microsoft's versions of Windows on the other hand were compatible to MS-DOS, were improved rapidly and with the third major release in 1990 finally became a commercial success. The reason for success clearly was the fact that MS-DOS and it's huge base of application software could still be used while the graphical user interface of Windows was modern, more convenient to use and more efficient.

Another success story apart from the market of operating system was Microsoft's developments of so called productivity applications like Word and Excel. In contrast to it's success in operating systems Microsoft did not have any first mover advantage, instead there were already strong incumbents (Lotus 1-2-3, WordPerfect, dBase) in the market when developments of those products were started (1980). Although some of the products were launched in 1982 and lot of money was put into their development the productivity applications did not become a commercial success initially as the incumbents were too strong and their products very well established. The very interesting strategic decision for Microsoft then was to switch the platform: It decided to release and improve the products not for the IBM but for the Apple platform, gaining experience and a dominant market position in the latter while still providing the dominant operating system for the former. The success then came in the late 1980s and 1990 at the same time as the first success with Windows. For those platforms Microsoft had a first mover advantage and was also able to integrate the software very well into the Windows user interface. What helped even more was that the incumbents still believed that IBM's OS/2 would become the platform of the future such that they did not put enough resources in the development of Windows versions of their products. When they finally were able to bring around a version for Windows 3.0 Microsoft was already in a great position and won a huge market share, for example a share of 70% in spreadsheet applications or 90% in word processor application by 1995. The success was even enhanced by Microsoft in 1990 with a new marketing strategy: Instead of selling every product at once, Microsoft also sold the package including every productivity application at a price that was very close to what a buyer then had to pay for one rival product alone. Although the competitors also had complementary products to their dominant ones, the integration of the whole

Microsoft package was better than what the others could offer. Microsoft came out as a winner in the fierce price competition that followed and by 1995 had a position of being dominant in both operating systems for personal computers as well as productivity applications leading to a share of 50% in the whole personal computer software market with a revenue of $6.08 billion.

4.1.2. More Success in Core Markets

The developments of Microsoft's products of course were never finished and therefore an array of new products were released: Windows 3.1 in 1992, later Windows 95 and Windows 98 and then Windows ME in 2000, all still relying on and compatible to MS-DOS. In 1993 Microsoft opened a new branch of development of operating systems by introducing Windows NT 3.1, an operating system with the same user interface as Windows 3.1, with no underlying MS-DOS but a newly developed kernel[6]. The original idea was to still sell the MS-DOS based Windows versions for private use and the Windows NT series for business, meaning workstation computers and servers. It included many new features including higher reliability, portability more platforms, not only IBM PC), a client-server concept, a DOS virtual machine (such that DOS software could be run) and many more (Operating System Documentation Project, n.d.). Successors were Windows NT 3.5, Windows NT 3.5.1, Windows NT 4.0 and Windows 2000. As DOS was not as important any more and the NT line offered the more modern product, Microsoft then decided to re-merge it's development and base it all on Windows NT. Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 followed. The latest version is Windows 7 which came out in 2008. Although Vista was not very successful and Linux operating systems were becoming better and better Microsoft's position was not challenged. Windows 7 seems to be highly successful again although it is still too early to measure the overall success. One of the many reasons for the ongoing success is that the only choice a buyer of a new PC usually has is between different Microsoft operating systems. Only very few vendors offer

[6] A kernel is the core part of an operating system.

PCs with a Linux system or without any pre-installed operating system[7].

Microsoft Office also was renewed and improved constantly leading to new releases every two years on average. Often times new products were introduced into the packages, the first one consisting of just 5 (Word, Excel, Access, PowerPoint and Mail) while newer version also included Outlook (instead of Mail), Publisher and some less known products. Microsoft still sold it's packages on Apples Mac platform although it is threatened lately by Apples own products. As in the operating systems market the only real competitor is an open source product. OpenOffice is based on Sun's StarOffice which was released as open source in 1999 (OpenOffice.org, n.d.).

4.1.3. Other Operating Systems

In 1992 Microsoft launched the WinPad project, a project that was designated to radically change the way an operating system was used with an eye on the mobile world (PDAs - personal digital assistants). The project failed in 1994 due to the fact that hardware for example for handwriting recognition or battery power for mobile use were not good enough at that time (HPC:Factor, n.d.). The Pulsar project was started instead with the aim to build a handheld device where Microsoft thought the future of computing would go but this also was a failure. The two teams nevertheless were brought together to form the Pegasus team to further develop their ideas which led to the release of Windows CE 1.0 in 1996, an operating system for handheld PCs that had pretty much the same look as Windows 95. The introduction of Windows CE 2.0 in 1997 made the system available on many different processors and for many more different purposes like automotive applications or consumer electronics, building certain features on top of the Windows CE system (like Windows CE for Automotive, Windows CE Pocket PC, Windows CE Smartphone). The system was also usually equipped with mobile versions of the Internet Explorer and Office applications like Word or Excel as

[7] Checking the product sites of the major PC companies like HP, Dell, Acer, Toshiba, Sony or Samsung shows that almost all of the PCs comes with a pre-installed Windows version

well as a connector to Outlook. This made Windows CE very compatible to the other Windows products on PCs and was able to maintain a good user experience. Many more version followed up to Windows CE 6.0, with Windows Mobile versions released up to Windows Mobile 6.5. The newest release Windows Phone 7 is based on Windows CE 6.0 and is redesigned to challenge new developments by competitors Apple (iPhone), Google (Android), RIM and Symbian (Bright, n.d.). It is not easy to get data about the success of the mobile and embedded branch of Microsoft's operating systems due to the numerous ways the systems are used since the beginning. One can still note that it became successful in the PDA market as it was able to become the leader in 2004 (Jaques, 2004) although the division still made losses of $64 million that year (HPC:Factor, n.d.). In the relatively new but growing smartphone market Microsoft's systems were not really successful, having market shares just around 15% from 2007 to 2009 even before Google Android smartphones were widely available (The GigaOM Network, 2010). Whether the Windows Phone 7 system can improve Microsoft's position in that market is yet to be seen. However, an alliance formed in 2011 with the biggest yet struggling mobile phone company Nokia might help in the long run (Microsoft, 2011).

4.1.4. Microsoft and the Internet

The Microsoft Internet Explorer was first released in 1995 and was built on source code from a company called Spyglass which was a young firm that came into the browser market a few years earlier (Sink, 2003). It is clear today that Microsoft did not foresee the future importance of the Internet and the threat that it could become to it's platform dominance: In principle, web browsers could become the basis of applications, making operating systems less important. This is a trend that we clearly see today (see also the section about Google) but which was of course not obvious in the early 1990s. When Microsoft realized the potential threat it was quick to buy the needed technology and put a big project team to make up for it. The first version of the browser was offered either as an add-on to existing Windows 95 installations and also was added to later distributions of that software. Microsoft also released version 2.0 later that year and made it available to download for free (Schnoll, n.d.). The dominant browser at that time (Netscape Navigator) was sold at around $50 but still Microsoft was not able to become very popular so version 3.0 was released already one year later and became a lot more accepted. With Internet Explorer 3.0 Microsoft also started to integrate the product very closely to it's operating systems, changing the system when the browser was installed. This practice was also done and improved when Microsoft introduced version 4.0 in 1997 which became an immediate success. Microsoft's share in the browser market was growing further and further with the releases of versions 5.0 (1999) and 6.0 (2001), peaking at around 95% in the US and Europe in 2004[8]. The market share then fell considerably due to the emergence of strong competitors like Mozilla Firefox (an open source product based on the Netscape Navigator), Opera, Apple Safari and Google Chrome and the refusal of Microsoft to improve it's product for some years. Internet Explorer 7 was released in 2006 and 8 in 2008 with many new features but both were not able to stop the losses in market share by Microsoft which in 2010 was down to about 60%. Version 9 was announced in 2010 and released in 2011 (IEBlog, n.d.).

Another important step was the introduction of MSN ("The Microsoft Network") in 1995 (Blogs, 2000) with an Internet search engine and as an ISP (Internet Service Provider) for dial-up networks. Today, MSN is mostly recognized as being an Internet portal and news site and many other applications have been added to the brand over the years. In 1997 Microsoft bought Hotmail, a startup that provided a free webmail service and had already millions of customers by 1997 and renamed it to MSN Hotmail (now: Windows Live Hotmail) (Craddock, n.d.). Another important application is MSN Messenger (now: Windows Live Messenger) which was launched in 1999 and is an application to for Internet chatting (Kunins, 2010). In 2005 Microsoft also launched it's own search engine and dumped Yahoo's advertising program from it's site in 2006 to do advertising on it's own. In 2009 Bing was introduced as a new and innovative search engine and in the same year Microsoft and Yahoo agreed to a deal such that Yahoo's web sites use the Bing

[8] One can find an excellent article about the problems of measuring market share in the browser market as well as a very comprehensive set of data sources at Wikipedia (Wikipedia, n.d.)

search engine (BBC News, 2009a).

4.1.5. Complementary Products and New Markets

Microsoft shipped it's first Media Player together with Windows 3.0 but this first version was not a very sophisticated piece of software, just able to play, stop and forward (Liron, n.d.). But as with other products it improved the player with every new version and provided it for free. From version 6.1 onwards Microsoft called it's media player Windows Media Player and with version 7.1 Microsoft also introduced it's own audio and video codecs (encoder and decoder). The latest version is Windows Media Player 12 which was introduced together with Windows 7. The work done in this field also led to special versions of it's operating system called Windows Media Center which was available starting with Windows XP for living room and other multimedia computers.

To develop software on it's operating systems Microsoft from the beginning included tools to help developers, mostly for the BASIC programming language (Lacher, n.d.; Microsoft .NET Support Team, 2009). It introduced Visual Basic in 1991 such that developers were able to use Windows' graphical elements to draw the user interface and use Microsoft's API. In 1998 this led to the introduction of the Microsoft Visual Studio, an integrated environment for Visual Basic but also many other programming languages and better support for developers, making it easier to write software for Windows. The .NET platform which was first introduced in 2002 developed this approach further and was put as a layer in between Windows and applications to provide interoperability between different Windows versions, an enhanced API, web technologies and a further convergence of the different languages. .NET therefore has similar goals as Java although limited to the Windows platform (while Java can run on any operating system). The Mono Project is an open source project that wants to bring the API and .NET applications to Linux and other operating systems although it usually lags behind Microsoft's versions of .NET (Mono, n.d.). Microsoft also introduced Silverlight in 2007 as a way to develop so called Rich Internet Applications (RIAs). Those applications run in a browser with a certain plugin and are usually written by using Adobe Flash technology or Java (Keesari, 2010). Again, Microsoft's product is not really interoperable as there are no plugings for some operating systems (most notably Linux) and certain browsers. The Mono Project also releases an open source version of Silverlight called Moonlight. The Visual Studio and .NET (latest releases are versions 10 and 4.0) are of course the main ways to write software for Windows even though the Visual Studio can be quite expensive depending on the package and the license. Whether Silverlight is able to take away a decent market share from Flash has yet to be seen.

DirectX is Microsoft's API for game development on PCs to allow easy and efficient access to the graphics hardware, sound hardware, gaming devices and so on. It was released in 1995 to make game developers who were used to programming games for DOS use Windows 95 for their new games (Coding Unit, n.d.a). From then on almost every year a new version of DirectX was released (for free) and producers of graphics hardware made their cards support the newest DirectX platform. DirectX is only available on Windows operating systems. OpenGL is a similar system that was created as an open platform in 1992 out of a graphics API by SGI (Coding Unit, n.d.b). It can be used on almost any operating system and hardware and is not only used for computer games but also by CAD programs and other professional software. It is interesting to note that Microsoft also participated in the development of OpenGL but dropped out later (seemingly 2003 but I was not able to find a reliable source).

Microsoft is also a player in the hardware business going back to the 1980s when the company developed a hardware mouse for the IBM platform which it still does (as well as keyboards). Other hardware products are webcams, headsets, gaming hardware and notebook accessories. A step into a completely new direction came when Microsoft launched the Xbox, a gaming console, in 2001 (thegameconsole.com, n.d.). Although Microsoft wrote software and tools for gaming consoles earlier, the move was seen as a surprise. One year later it introduced XBox Live as a network for gamers that could purchase and play games online against other

XBox players. In 2005 a new hardware version was released, the XBox 360. Although both consoles were not extremely successful (rankings 14 and 10 in all-time total gaming hardware sales) they seemed to have challenged incumbents Sony and Nintendo to release newer version of their platforms (PS3 and Wii) and also provide online gaming networks and newer technology (VGChartz, n.d.).

The last product group I want to mention here are mobile devices, products that are seen as a way to battle Apple's success in the mobile world. In 2006 Microsoft introduced the Zune player, an MP3 player and it was "predicted that the Zune could do to the iPod what Windows PCs did to the Apple Mac“ (Robins, 2009). This did not hold true however as Apple's iPod sold up to 50 times more than Zune until 2008 (same source). Still, Microsoft is upgrading, developing and selling Zune players and was even building a Zune network with a PC software to connect Zune to, an online music and video store and software for other products like XBox and Kin. Kin, a project to also offer phones directly (and not only operating systems and software) was stopped just months after it was introduced in May 2010, so this was hardly a success (Foley, 2010).

4.1.6. Dominance, Antitrust and Other Legal Issues

The first important legal issue for Microsoft was one posed by Apple which sued Microsoft in 1988 about copyright infringements of the Windows GUI but Microsoft won in 1994. More information about this case can be found in the section about Apple.

In 1993 the US Justice Department started an investigation about Microsoft's marketing practices for it's operating systems, taking over from the Federal Trade Commission because of deadlocks in it's decision process[9]. This started a massive process about several issues that would last for about 10 years between the Justice Department and Microsoft, a summarizing timeline can be found at Wired.com

[9] The FTC Commission voted 2-2 which stopped the investigation in 1993 (Gilbert, 1999).

(2002) and a number of official documents about the case can be retrieved from the Internet (Department of Justice - Antitrust Devision, n.d.). These first investigations were settled rather quickly, leading to a consent decree that Microsoft signed in 1994 (approved in court in 1995) and which forced Microsoft into giving up some of it's anticompetitive practices. The European Commission signed a similar deal with Microsoft the same year. Gilbert (1999) did an economic analysis about the case and found several issues with Microsoft's marketing behavior from a competition policy standpoint:

- Microsoft sold long-term contracts with large minimum commitments to OEMs (Original Equipment Manufacturer) which increased barriers to entry for possible competitors even for higher than competitive prices. Of course one can ask why OEMs would do such a contract that prevents entry from a competitor which is usually good for OEMs. Gilbert's reason comes from the fact that every individual OEM is too small to make entry from a Microsoft competitor profitable and therefore is better off signing the long-term contract. The high minimum commitments serve as a way to leave just very little residual demand for entrants. Gilbert shows that many contracts were very long (three to five years) while the mimimum commitment often exceeded 50% of expected sales of one OEM. Of course, also positive effects can be found in these contract which can be summarized by higher security for both the buyer and the seller.

- Microsoft sold different licenses and one of those was a so called "per processor license“. With such a license an OEM had to pay a fee to Microsoft when it sold a product with a certain processor independent of whether the system was shipped with a Microsoft operating system or not. Microsoft usually sold this license at the lowest price and the share of licenses reached 60% in 1993. This also deters entry because an entrant has to sell at a very low price to make up for the fee the OEM has to pay to Microsoft anyways. Even a more efficient firm might have problems entering the market in this setup.

- Microsoft also imposed excessive nondisclosure agreements (NDAs) on de-

> velopers they worked with during development of a new operating system. For developers it is important to be able to offer applications right when the new operating system comes onto the market. For Microsoft the agreement is important because otherwise developers could sell information about the new product to competitors. However, the agreement for the development of Windows 95 was so strict that it prevented some developers from working with competitors for more than a year and therefore encouraged developers to develop just for the Microsoft operating systems.

The consent decree limited the contracts to a maximum of one year with no minimum commitment and limited the duration of NDAs. Microsoft was also prohibited from selling per processor licenses although price discrimination and volume pricing incentives were still allowed. It also prohibited Microsoft to tie it's operating system sales to other products as well as contracts that restrict OEMs to purchase products from competitors.

Complaints by Microsoft's competitors in application software that Microsoft used it's internal knowledge about their operating system to gain a head start in application programming for it's system were not discussed in the case although Gilbert acknowledges it to be a (potentially) big problem and even Judge Sporkin raised the issue during the hearing for the consent decree[10].

In 1997 the Justice Department again started investigations because they believed that Microsoft violated the agreement by tying operating system sales to it's Internet Explorer products, requiring OEMs to install the Internet Explorer on sold computers with a Microsoft 95 operating system. The Justice Department demanded a fine of $1 million per day of violation and later even a break up of Microsoft into two separate companies: Operating systems and application software. Microsoft's argument was that the Internet Explorer was not a separate product but integrated into the operating system and that the Internet Explorer improves the latter. Fisher (2000) argues that this is not necessarily the case and Microsoft's

[10] Note that also Campbell-Kelly argued that it was important for Microsoft to be able to bring the Office applications for Windows 3.0 to the market first.

action only made sense taking into account the fact that the Netscape browser was a long-term threat to it's monopoly profits[11]. Davis and Murphy (2000) argue that versions of Internet Explorer have been shipped independently but they also show possible welfare enhancing effects and a zero-price strategy that is not an anticompetitive strategy per se but makes sense for Microsoft nevertheless.

The trial against Microsoft lasted several more years with inputs from many companies and questionaires of Microsoft's leaders. In 2000 the court ruled that Microsoft had "a monopoly in the market for Intel-compatible personal computer operating systems" due to the "application barrier to entry". This barrier prevents entry into the market because consumers would not switch to a new operating system when there was just a limited number of applications available and software companies would not develop applications for an operating system with just a few users. Furthermore, it was acknowledged that middleware was a potential threat to this monopoly because it could provide a layer between applications and the operating system and therefore erode the application barrier to entry. The court found Microsoft guilty of anticompetitive behavior against producers of middleware (like the Netscape Navigator web browser or Java) and ordered a breakup of Microsoft's operating systems and applications business. Microsoft appealed and in 2001 the breakup was reversed although the finding of monopoly power and anticompetitive behavior was upheld, leading to a a settlement between Microsoft and the Justice Department in 2002. This settlement orders several rules of conduct for Microsoft including it's licensing practices and it's dealing with OEMs especially when it comes to middleware. It was further agreed that Microsoft had to make available APIs, communication protocols and related documentation "on reasonable and non-discriminatory terms" to third parties and make it possible for users and OEMs to uninstall Microsoft's middleware and designate an alternative middleware in place of Microsoft's.

[11] He also argues that there has been more anticompetitive behavior by Microsoft in the direction of Java by offering a Java environment ("polluted Java") that was incompatible to others and Apple by threatening to not develop Office for it's platform anymore and put resources to compete against Apple in audio and video. Apple also used the Internet Explorer as it's main browser maybe due to these threats.

Microsoft's issues with it's strategies and behavior took a different turn in Europe: Starting points were complaints by Novell about Microsoft's licensing practices in 1993 and Sun's complaints that Microsoft did not provide information about it's network interfaces and therefore prevented competitors from developing a compatible network operating system (Abu-Haidar, 1993; McCullagh, 2002). This led to investigations by the European Commission which found that Microsoft withheld necessary information from it's competitors and bundled it's operating system with the Windows Media Player (Parsons and Best, 2004). The EC imposed a record fine of €497 million and required Microsoft to offer a version of Microsoft Windows without the Windows Media Player and open up information about it's interfaces for competitors[12]. The fine was further increased in 2006 as the European Commission was unhappy with Microsoft not providing sufficient technical information (Lawsky and Zawadzki, 2006). In 2007 the Court of First Instance dismissed most of Microsoft's appeal to the ruling including the fine, the requirement for a media player free Windows and the requirement to provide information to competitors (Court of Justice of the European Union, 2007). In 2008 an even greater fine (€899 million) was imposed by the European Commission which was also appealed by Microsoft while the European Commission started to scrutinize on Microsoft's bundling of it's browser with Windows (BBC News, 2009b). An agreement about this issue was made in early 2010 when Microsoft agreed to offer it's customers in Europe the choice between 12 different browsers including Microsoft's Internet Explorer (BBC News, 2010a). In 2005 South Korea's Federal Trade Commission imposed a fine of $ 23 million and ordered that Microsoft had to offer a version of Windows without Windows Media Player and MSN Messenger. An appeal by Microsoft was turned down in 2006 (Mook, 2006).

4.1.7. Summary, Strategies and Openness

Microsoft's root of success is closely linked to the success of the IBM platform, a success that many authors believe is independent of the quality of Microsoft's

[12] A much more detailed impression about the decision can be gained by the issued commission decision (Commission of the European Communities, 2002).

operating systems. The success of Microsoft in the beginning is therefore not only a matter of strategy (being the first and cheapest operating system on the platform) but also of luck, given the future importance of the platform that could not have been foreseen and the choice of IBM to license operating systems instead of providing it's own. The success in productivity applications can be linked to the success of Windows 3.0, the head-start it got and the inability of the former incumbents to provide timely products for Windows 3.0 and to counter Microsoft's packaging strategy. Both products still make up a huge portion of Microsoft's revenues today.

The platform Microsoft can provide is not a typical monopoly but one that is protected even stronger: Due to lock-in and learning effects of users but also due to the application barrier to entry. The only potential threat to this huge market power lies in the possibility that operating systems and productivity platforms might not be needed as much in the future: Middleware like web browsers might provide the basis to (Internet) applications of the future, taking down the barrier to entry. More recent developments of mobile platforms or the convergence of computers and multimedia systems might require different operating system capabilities, practically lowering the lock-in and learning effects that we observe. Microsoft's strategy is to have a foot in the door on those developments by providing it's own web browser, mobile and embedded operating systems, multimedia and mobile devices and gaming consoles. The impression though is that Microsoft's overall success in those areas is not as good as in the core business: Internet Explorer developments have been stopped (and picked up again just very recently) leading to big losses in market share. The new mobile devices and operating systems are not receiving the same hype as competitor's products and it's gaming devices sales are still behind those of the incumbents. In fact, Microsoft is not getting the lead in those developments but is usually behind facing a tough leader. The operating system market power that helped gain momentum in some areas cannot be used anymore due to the consistent pressure from competition agencies around the globe where especially the European Commission is forcing Microsoft to open up to competitors.

The strategies of Microsoft concering openness are mixed: In the beginning, Microsoft developed for several platforms, even developed it's own Unix operating system that could run all DOS applications and developed applications and programming languages for several platforms including Apple and other Unix systems. Microsoft's strategy to focus on IBM's platform and by providing it's DOS operating system to many vendors that offered IBM-type personal computers allowed them to set a (proprietary) standard in the industry that it still holds. Although Microsoft always offered products such that other companies could provide application software for it's operating system (a necessity), the inner workings of Microsoft's operating systems have not been available to other companies which might have been an advantage in developing complementary products (Microsoft Office products or the Internet Explorer) and also helped strengthen the entry barrier for prospective competitors. It needed several trials in the US and Europe to make Microsoft open up some of it's interfaces, for example the SMB protocol that connects Windows PCs or the MS Office file formats.

On the other hand Microsoft often tried to squash open standards and build up it's own: W3C is an initiative that provides open standards for HTML and JavaScript to develop web pages. Microsoft's browsers were always known for not complying with the whole set of standards and even introduced a JScript language which was basically an extension to JavaScript not approved by the W3C. Another technology is ActiveX that allows software to run in a browser and behave more like local software than typical web pages. The ActiveX technology is not only known for many security bugs but also for the fact that it could only run on Microsoft's browsers. DirectX is also a proprietary 3D standard developed by Microsoft although with OpenGL an open alternative is available.

A very interesting aspect in Microsoft's strategy about openness and open source software in particular are the so called "Halloween Documents". Those documents were written by a Microsoft product manager and posted on the Internet by a well known Linux developer, although it is not entirely clear how he got it in the first place[13]. The name "Halloween Documents" was given due to the fact that the

[13] People at Microsoft acknowledged that the documents were authentic although they referred

release date of the first document was on November 1, 1998. A news report about the documents can be found at (Trott, 1998) while the original and commented release can be found at (Raymond, n.d.). Those documents are studies about the open source process and success as well as Linux and how it relates to Microsoft's business, products and strategies. The basic result of those investigations is that open source and Linux in particular can be a threat and that the quality of Linux cannot be neglected. Open source is strongest when protocols are "commodities" (my take would be to use the word "open"), therefore the authors argue: "By extending these protocols and developing new protocols, we can deny OSS projects entry into the market." Although it is not clear whether Microsoft really developed these insights into a real strategy, Microsoft's behavior with respect to open standards and interfaces seems quite close.

Another point made by the authors of the papers is that from Microsoft's point of view Linux does or will "cream skim" the best features of Microsoft's operating systems and include it into it's own features. This would be a big threat to Microsoft as it has to invest into the development of the features which then would be copied into Linux. The authors also recommend that Microsoft should investigate into whether patents and copyright might be a way to compete against Linux. In 2007 Microsoft claimed that Linux and other open source software infringes a total of 235 patents held by Microsoft although the company later promised that they will not sue Linux users (Parloff, 2007; McDougall, 2007). Microsoft would not reveal which patents were infringed, leading the Linux founder and main developer Linus Torvalds to tell Microsoft to test the allegations in court or at least open up the information so that Linux could develop around those patents (Babcock, 2007).

Still, more recent developments seem to go into a more open direction, maybe forced by the legal problems with the old strategy, maybe due to a change in culture inside the company or maybe due to demands for more open and interoperable products. As already mentioned above Microsoft does not prevent open source

to it as "low-level engineering studies" while the developer believes it reflects Microsoft's strategy against open source.

developers form porting it’s .NET and Silverlight framework to Linux but even seems to encourage it at least to a certain extent. In 2006 Microsoft and Novell (the company distributing the popular SuSE Linux distribution) announced a deal that included a patent agreement, an agreement on interoperability (in virtualization, web services and data formats) and a business collaboration agreement (Gardner, 2006). This agreement not only freed up Novell from any patent issues with Microsoft but also made Microsoft an indirect distributor of Linux, buying 70,000 coupons for SuSE Linux for sale to it’s customers. In 2009 an agreement with Red Hat (distributor of Red Hat Linux) was announced to work together in the area of virtualization (Vaughan-Nichols, 2009). With Microsoft Office 2007 Microsoft introduced a new open and standardized document format called Office Open XML[14]. Although Microsoft therefore added another standard to the already existing open and standardized Open Document Format (ODF) used by OpenOffice Microsoft also developed a plug-in such that ODF could be used with Microsoft Office. Later products also supported Adobe’s PDF format. Microsoft now even has it’s own branch of open source products, two different open source licenses for those products, an open repository for open source products (CodePlex) and claims that it is working with and improving existing open source projects like the Linux Kernel, Apache or PHP[15]. Critics however claim that Microsoft’s step into open source development is not sincere and that those projects are “encumbered with a dependency” on Microsoft’s baseline products (Oliver, 2010).

Overall, one can claim that Microsoft faired quite will in the past with it’s closed approach. It allowed them to gain dominance in some sectors and gave them a starting point for the jump onto others. Competitors of course could not survive when Microsoft included products into it’s operating systems (seemingly for free or even enforcing it’s use by OEMs). Some of those strategies were decided to be anticompetitive by courts in the US and Europe and Microsoft was forced to give up some of it’s practices leading to a more open approach in the recent past. Of course, there is still no conclusive evidence how Microsoft’s strategy and it’s use of

[14] Although it was argued that it might not be completely documented and free of patents.

[15] The Linux Kernel is the main part of Linux, Apache is the leading HTTP web server and PHP is the leading scripting language for dynamic web pages

openness will look like in the future, especially since it is not clear yet how much importance operating systems will have when mobile devices, virtualization and the convergence of products become even more important. And of course there is always the threat of some middleware to take the barrier to entry away and the always-continuing evolution of Linux or OpenOffice as alternatives to Microsoft's main products.

4.2. Apple

4.2.1. The Beginning of Apple Computer and First Success

Apple Computer was founded by Steve Jobs, Steve Wozniak and Ronald Wayne in 1976, the latter dropping out of the company just months later due to the risky nature of the beginning business. Jobs was a college dropout who was working at Atari at the time while Wozniak (also a college dropout) was working at HP. The two were friends for many years and shared an interest in electronics. The first computer (later called the Apple I) was developed a year earlier by Wozniak who was driven by the idea to build a computer on his own and was inspired by the Altair 8800 computer, a kit to build a personal computer. The Apple I was quite basic with cheap components and was maybe more a study or a project of interest for Wozniak but Jobs saw the opportunity to make it a product[16]. This also clearly showed the different type of people the two founders were: Wozniak was an engineer, some even believe an electronics genius who never wanted to have his own company and also was no visionary but driven by interesting projects or difficult problems in engineering. Jobs on the other hand always had big ideas, especially for business but was maybe just an average engineer. After neither HP nor Atari wanted to use Wozniak's development Jobs was able to persuade Wozniak into forming a company and brought Wayne over from Atari. They sold the Apple I to local computer enthusiasts, stores and the Byte Shop possibly the first retail

[16] It was not really a computer in a modern sense but rather a circuit board to which a case, peripherals and a power supply had to be added to make it useful.

computer store chain. This allowed them to make a solid profit and soon Wozniak began working on an improved computer called Apple II while Jobs worked on getting enough funding for increasing their business.

They got to know Mike Markkula, a 34 year old retiree from Fairchild Semiconductor who came out of retirement to work on a business plan for Apple and provided funding from his own pocket as well as from external sources. Markkula, Jobs and Wozniak incorporated Apple in 1977 and purchased the old partnership that Wayne had already left. The Apple II came out in April 1977 and was sold at $1,298 (the Apple I was sold at $666). It was not only revolutionary in it's features but also was the first computer for the mass market, with a plastic case, an integrated keyboard and power supply, color graphics capability (one could use a TV set as a display), 8 expansion slots for additional hardware and a Basic programming language licensed by Microsoft called Applesoft. Important for it's success was also a disk drive developed by Wozniak that could be used in an expansion slot but even more important was a program called VisiCalc, a spreadsheet program written in Applesoft that became very popular. The Apple II and successive versions of it were very successful and sold over 300,000 times before IBM even introduced it's first PC in 1981. What proved very popular were the expansion slots of the Apple II and many other companies sold hardware for it. Even after Apple introduced several other products (see below) and wanted to drive demand to those more expensive ones the Apple II still held it's own and was Apple's cashcow for a long time. The last version of Apple II called Apple IIGS was introduced in 1985 and sold until 1994.

Apple went public at the end of the year 1980 and made the founders and many of the first employees instant millionaires. After that the companies' fortunes turned a little bit as Wozniak was heavily injured during a plane crash and left the company for a couple of years. By 1981 it became clear that the developments of the new products did not go as smoothly as expected as many engineers worked on side projects with not much commercial sense and so some employees had to be laid off that year. The new products that were started in three different projects were constantly outsold by the newer versions of Apple II which were introduced in parallel. The first new project was the Apple III, a computer designed for business with more features while Apple II should still sell for private use. Apple III was first announced in May 1980 but due to problems with the design and production the computer was altered and was first shipped in March 1981. After some of the computers were sold in 1981 it also became clear that there were some principle problems with it, especially with the cooling mechanism. Apple IIIs therefore often failed during usage and were not very reliable. The price was also very high compared to the Apple II at around $4,000. Even after some of the technical problems were revised and the price was lowered to around $3,000 later in 1982 the Apple III was a disappointment for Apple, it did not nearly sell as well as the Apple II did and was dropped in 1984.

Even before Apple III was introduced Apple started another project called Lisa from which the company hoped to develop a computer to replace the Apple II (and I guess also the Apple III) line of products. The project was heavily influenced ("inspired" is the word used at Apple) by the Xerox Alto, a revolutionary computer developed at Xerox Parc in the 1970s that already had many features of modern computers like a graphical operating system with overlapping windows controlled by a mouse[17]. It was clear immediately to Jobs and others at Apple that this would be the future of computing and they wanted to build something similar with their Lisa project. Apple also hired many employees from Xerox during that time. When volume shipments of the Lisa computer began in June 1983 the price for the computer and a bundle of software was set to $9,995 and it had problems from the start. Next to the hefty price the Lisa was incompatible to the Apple II and III and therefore no other software was able to run on it. The Lisa also was too slow and unreliable and Apple had no experience in selling this kind of computers that were too expensive for the typical Apple user at that time. And then, it was clear to many customers that the computer from the third project (Mac) was to be introduced shortly, having almost Lisa-type features for a much lower price. Although Apple later unbundled software and hardware and sold the hardware at a much lower price the computer never became a success and the project was stopped in 1985.

[17] The Xerox Alto was never sold but it is believed that it would have cost around $40,000.

The third project was called the Macintosh and was started by former Xerox employee Jeff Raskin to build a computer at a very low price: The target at the beginning was around $500. Although Jobs was not a fan of the project initially he took it over after he was not allowed to work with the Lisa team and the project goal seemingly changed from making a cheap computer to beating the Lisa. The “Mac“ was introduced in January 1984 and was smaller and faster than the Lisa but also cheaper at $2,495 while many features like the graphical window manager and some software were the same. Still, the first Mac had some flaws, for example not much memory and also no color capabilities as well as no disk drive and no expansion slots, but it nevertheless sold much better than the Lisa initially. Although just half a year later Apple quadrupled the memory of the Mac the sales dropped late in 1984 and also failed the big expectations. For private use it was maybe still too early and too expensive to conquer the mass market and for business use it was just not designed for, having not enough memory and not enough (and possibly the wrong) applications that could run on it. What did not help either was that the original Mac development team was exhausted and many left the company or the project so further improvements of the Mac were not easy to come by.

4.2.2. The First Crisis

The problem for Apple was that it still had to rely on it’s old Apple II (although often enhanced) and the new projects although in some aspects revolutionary did not sell well. And the business environment got tougher: IBM introduced it’s PC in 1981 and was very successful immediately even when Apple believed that the technology was far inferior to it’s own. Other companies came into the market and offered product clones of the IBM PC and Microsoft was selling and licensing it’s DOS system to all those companies, hence, building up a network of compatible computers which also attracted software companies. Although it is not entirely clear how IBM was able to gain a great market share with a technically inferior product, there are some hints:

- The IBM PC was cheaper than all the new Apple products and about the same price as Apple II.
- IBM was the biggest computer company in the world, making revenues with mainframes, services and software. Therefore, there were big expectations for the product and it's further development.
- Many firms immediately started to develop software for the IBM PC, believing in the success while strengthening the platform.
- The platform was open in a way that competitors could offer compatible products ("clones“). This led to a diverse set of offers for different customers.
- As described above, Microsoft quickly became the dominant operating system for the IBM PC but it was also allowed by the licensing agreement with IBM to license it to other vendors, therefore enabling a bigger platform and high compatibility for software companies.

As the IBM platform took off, often driven by new Intel processors, new versions of DOS and better and more application software, Apple was stuck with revolutionary but overpriced products that had technical flaws and not enough software to compete and an old product that at Apple many executives did not really like anymore (even though it was still the cash cow). The introduction of the MacIntosh Office in January 1985 was also unable to turn the tide as one crucial piece (a fileserver) would still need two years onto the market. Apple also posted it's first quarterly loss and the stock fell to an all-time low in June. And what was also pretty clear in the midst of the 1980s: Computers or computer hardware became a commodity market. Even IBM had problems in the PC market competing with ever decreasing prices against other companies on it's own platform. The important piece was software and the operating system was important as an underlying platform while providing programming languages and libraries for software development. The hardware market slumped during that time which obviously worsened Apple's problems.

The solution might have been to separate hardware and the operating system which basically meant opening the operating system and trying to make it a standard in the PC world (even on the IBM platform). This would have been possible for Apple in a way Microsoft did it in the 1990s because Apple clearly had the better product. When Microsoft introduced Windows, it needed until version 3.0 in the early 1990s to offer a quality (in terms of features) that Apple already achieved with the Lisa and the Mac product line. Microsoft even approached Apple in 1985, handing over a basic strategy paper to make Apple's operating system an industry standard[18] (Carlton, 1997). Today, it might seem strange that Microsoft did this but back then it was not clear whether Microsoft would ever reach the kind of quality and Bill Gates was often cited in praising the Mac operating system. Furthermore, Microsoft was making a lot of money with software for the Mac and believed it could make even more when the operating system became an industry standard. History did take a different route though as many core Apple developers and managers believed that the operating system was their asset and allowed them to sell hardware: "Why sell an operating system software for $100 like Gates did when you could wrap it in hardware and sell it for $2,000?" Goodell (1996, p.159). Furthermore, Apple's philosophy was (and is) to build a complete product with software tailored to the hardware and vice versa. This is what made the products easier to use and less error-prone than in the IBM-Intel-Microsoft world. Although Apple often discussed licensing the operating system later, when they finally agreed to do it, it was already too late (see below).

From a personal perspective also many things changed that year: John Scully, the CEO who was hired personally by Steve Jobs in 1983 fell out with Jobs who on the other hand could not stand the criticism of the MacIntosh. In the struggle for power Scully stripped Jobs who had still a share of about 11% of all operational duties. Although leading managers wanted to keep him in as chairman, Jobs resigned and started NeXT with several other former Apple employees. Earlier that year also Steve Wozniak resigned citing frustration over the lack of support from the management for the Apple II line of computer he was still (or, again) working on.

[18] Reading this, one can clearly identify the strategy that Microsoft chose when Windows was finally good enough to do it.

Therefore, in 1985 the founders of Apple were no longer part of the company. Also, about a fifth of all employees at Apple were released that year.

What got Apple out of the slump was not something that Apple did but was coming from the outside: Aldus, a very little startup software company released the PageMaker in July 1985, a software to put text and graphics together on documents to print them out. When Apple found out about that software they donated a Mac such that Aldus would write the software for it and it became also clear that together with a Mac and Apple's laser printer (a product initially developed for the MacIntosh Office campaign) it allowed publishing from a home computer and therefore became a huge success. Another killer software was found (as VisiCalc was for the Apple II) and together with it's printers Apple was able to turn the Mac into a successful product and the company back on track. Lots of new Macs with better hardware and upgraded software were released, the hardware was finally opened for expansion slots and together with Aldus PageMake, Microsoft's productivity software, Apple's great operating system and solid hardware it was able to offer a lot to potential customers with profit markups of 50% or even more.

4.2.3. The Second Crisis

But the second crisis was already in it's beginning stage although Apple was again a hugely profitable company for many years. Although the IBM platform with Microsoft DOS and a lot of application software outsold the Apple products easily Apple had a nice niche and lots of profits. The problem so to speak was Microsoft. Although it was also a key partner for Apple from the very beginning, providing programming languages and software for the Apple II and later for the Mac, it was also clear that as the dominating operating system producer for the IBM platform it was also a competitor. In the mid 1980s Windows was still no competition though but it was clear that many of it's features were similar to those of Apple's operating system. Although Apple was the much bigger company in 1985 and

Apple threatened to sue for copyright infringement Bill Gates was able to get Apple into an agreement that allowed Microsoft to use some of Apple's technology for a delayed release of Microsoft Excel for Windows. Clearly, Apple needed Microsoft in 1985 more than Microsoft needed Apple but the deal was a bad one for Apple especially as Microsoft wasn't even able to ship it's product for Windows as early as Apple feared.

That the scope of this agreement was not entirely clear became obvious when Apple started a copyright infringement lawsuit against Microsoft (and HP) after Microsoft released Windows 2.03, coming closer and closer to cloning Apple's user interface. The question then was whether the agreement allowed the use of Apple's technology just for Windows 1.0 (as Apple was claiming) or for all subsequent versions as well (as Microsoft was claiming). A rule in 1989 declared that almost all of Apple's violation claims were allowed due to the 1985 agreement. Later, the courts favored Apple's stance a little more but in 1993 the case was decided in favor of Microsoft. Further appeals by Apple were dismissed. Another major point made during the trial was that even when some of the stuff Microsoft was using was not in the scope of the 1985 agreement, those objections were just not protected by copyright law anyway, leaving Apple with not much legal power.

It was therefore clear for Apple that Microsoft was allowed to build an operating system close to the Mac operating system and that it will eventually catch up and take away Apple's technological lead. Therefore, Apple went back to what it always did when facing competition in it's core market: Build new revolutionary devices and open up a new market which was believed to be the future. Apple's answer was a project called Newton, already started in 1987 as a research project for an eventual Apple handheld computer. The project apparently became a favorite of CEO Sculley and therefore was pushed, setting the initial release year to 1992. Most important targeted features were handwriting recognition, a new programming language, productivity software and long battery life. But the development just did not work very well as the focus was often shifted between different versions (or sizes) of the product and the processor had to be changed during the development, setting back the efforts several times. And the announcements of the product by Apple's managers and the huge expectations did not help either while ship dates were missed frequently. And when the first product, the Newton MessagePad, began shipping in August 1993 at a cost of $699 the product had still many bugs[19] and the handwriting recognition was very poor. There was also no third party software for the product. Still, the Newton received lots of positive feedback and was again a revolutionary Apple product but just did not work as well as expected and maybe the time also was not right. Companies like Palm or Sharp would eventually take over the market with cheaper and less error-prone products. And even Microsoft stepped into that business by introducing it's Windows CE operating system in 1996. Apple still improved it's MessagePads and eventually sold a greater variety with lower and higher priced products but had to discontinue all development efforts in 1998. The installed base in 1998 was just about 150,000 to 300,000 while Windows CE already sold 500,000 times and the total development and marketing cost for Apple's Newton was estimated at about half a billion dollars.

Although Apple was able to land a hit with the PowerBook series, a set of laptop computers that started in 1991 and the MacIntosh computers were still selling well, problems began to mount at Apple in the midst of the 1990s: In 1994 and 1995 Apple completely misforecasted the demand for it's products, trying to sell high-price computer when customers wanted low-price and vice versa. The market also picked up in 1995 but Apple underestimated the demand and could not meet it. It also had troubles with it's suppliers such that in June 1995 there was a backorder of $ 1 billion in Apple products. Apple's newest PowerBook 5300 also had major design issues: First with it's battery that took fire in early releases and then with the case, the power plug and power supply, leading to two recalls and a loss of a third of PowerBook revenues in 1996.

When Michael Spindler became CEO from John Scully in 1993 he finally addressed the licensing issue. The first problem was that operating system and hardware were

[19] A "bug" is a design or programming error in a software. There are many stories about the origin of this term in computer science but it often has to do with a small animal destroying some electronics parts in ancient times of computer engineering.

so tightly coupled that there was just minimal interest from PC vendors because many believed they could not compete with Apple on their own platform. Apple therefore joined (together with Motorola) IBM's Common Hardware Reference Platform (CHRP) in November 1994 and this would later become the PowerPC platform, a hardware platform with standard ports and components that could run Apple's operating system. This attracted some smaller vendors to license Apple's operating system. Apple's goal was to get a market share of the platform up from about 10% to about 20% in five years but it did not work: Scepticism took over just one year later at Apple as one competitor (Power) was able to gain a 10% share on the platform by targeting the high-price segment. Apple's managers believed that they lose more to the competition than they gain from the expanded platform and the licensing fee. The experiment was finally stopped in 1997, marking the end to companies like Power. And then, Windows 95 came out, finally an operating system from Microsoft that was as good as Apple's which gained a huge momentum also due to Microsoft's marketing and advertising efforts.

4.2.4. Back to Success and New Products

Apple was in deep trouble and reported a loss of $740 million in the second quarter of 1996. 2,800 employees were laid off that year but one important employee came back to the company: Steve Jobs, who since he left 11 years ago has made a fortune with his investment in Pixar, came back due to the fact that Apple bought the company he founded: NeXT Computer Inc. Jobs was first put in an advisory role for then-CEO Gel Amelio but became the interim CEO in 1997 after Amelio resigned. The way Jobs reshaped and turned around Apple is widely viewed as the major reason Apple became a very successful company again.

The reason for Apple's purchase of NeXT was it's operating system. This has been an issue for many years for the people at Apple as they felt they needed to remake the Mac operating system to get back into the technological lead. Projects at Apple like Copland did not work out and the incremental improvements of the old operating system did not prevent Microsoft from catching up. When Apple fell into the deep crisis in 1996 and internal projects were still far from being ready for the market, Apple's management looked for external help: They talked about making BeOS it's new operating system or Suns Solaris and even thought about getting a license for Windows NT, Microsoft's successful server and workstation operating system.

The decision for NeXT in retrospect was the right one. Not just because of the operating system that would form the basis for future Apple operating systems but because it brought Jobs back to Apple who immediately made important decisions to get Apple back on track: Several projects and product developments (like the Newton) as well as the licensing experiment were stopped, new ways of marketing were explored (deals with CompUSA, an Internet store for Apple products and later even retail stores called Apple Store), a greater emphasis was placed on design and new products and product categories introduced. Jobs even made an agreement with Microsoft, ensuring continuous software development from Microsoft for the Apple platform and cash for making the Internet Explorer the standard browser for the Mac and patent agreements. Apple under Steve Jobs was to get back to what made them very profitable in the years before: Offering high quality and technologically advanced products that allowed them to get around price competition and cash in high margins.

Apple's first product developments after Jobs became the interim CEO were big successes: First, the high-end Power Mac with a G3 processor from IBM and Motorola sold very well and for a nice margin. Then, the iMac, a consumer PC with a new design including an integrated display and equipped with lot of great hardware for just $1,299 which was very popular especially for new PC users. And then, the iBook, a consumer laptop which also was a great piece of design and hardware for only $1,599. One year after it's introduction the iMac became the then best selling computer in the US and Apple doubled it's market share to 11.2% in 1999.

The operating system developments led to Mac OS X, a new operating system based on a Unix kernel called Darwin which was finally shipped in January 2002. It immediately became the standard operating systems for all Apple computers and it also had an emulator such that old Mac software could run on it, ensuring full compatibility to the old platform. Further versions of Mac OS X are still used today as the operating systems for Apple's personal computers. Although the technological lead is not as big as it was in 1984 when the MacIntosh was introduced one can still argue that Apple has the lead over Microsoft in the operating system department, even after the introduction of Windows 7[20].

The first profit after the crisis was made in the first fiscal quarter of 1998, leading to an annual profit of $309 million that year. After Apple made a profit of $601 million in 1999, Steve Jobs agreed to become the CEO, stripping the "interim" from his title. Although also Apple was hit by the burst of the dot-com bubble (leading to a loss of $25 million in 2001), the foundation was laid for Apple to open up new markets in the next century.

4.2.5. Digital Lifestyle

In 2001 Apple introduced iTunes, a software to play and deal with music files and burn them to CD. Later the same year Apple introduced it's first iPod and the second version of iTunes. The iPod was an MP3 player and although there were already many players on the market and it had a high price of $399 it became a breakthrough product for Apple. Not only did it present a good compromise between size and storage capacity, it also had a distinguishable design and integrated very well to the iTunes software that could transfer music to the iPod via a Firewire connection which at that time was a lot faster than USB. But these first steps into a new market were just the basis for many things to come. In 2002 models with higher capacity and slightly changed design were introduced and iTunes was also made available for Microsoft Windows.

[20] When Windows 7 came out there were again accusations that several features were copied from the Mac OS X although there are no legal disputes over that matter.

Another breakthrough was the introduction of the iTunes Music Store in 2003. Although there have been ideas and concepts before how to sell music over the Internet, nobody was able to make it work. Apple on the other hand did, by integrating it into it's iTunes software and making a great user interface, by getting all the five big music companies on board (such that over 200,000 songs were already available at the introduction) and by having a very convenient sales concept such that every song cost $0.99. And although Apple did not have a big direct profit selling music it won indirectly by selling even more high margin iPods who in 2004 was by far the best selling MP3 player. Further hardware version were introduced later including USB connections, smaller devices without display, video capabilities, cameras and so on.

An even greater and in the future maybe more important market was opened for Apple when Steve Jobs introduced the iPhone in 2007 even though there was already a huge saturated market for mobile phones (dominated by Nokia) and a smaller market for so called smartphones with strong incumbents (RIM, Palm and again Nokia). Smartphones like the iPhone are devices that combine the features of mobile phones with features of organizers (the old Newton was a device of that kind). One can call and write SMS, one can read, write and organize files, one can deal with an electronic calender, tasks, contacts and email, one can connect to the Internet to browse webpages and one can run software. Apple's iPhone can do all that but to distinguish itself Apple added features that made it a revolutionary product: Most other smartphones came with tiny keyboards or a little touch stick but with the included multi-touch panel one could use the Apple iPhone with finger touches and gestures only. This was a new way of control and user experience and became a huge success. Furthermore, the big and high resolution screen and the great integration of Apple's Safari web browser together with the touch control made it a great device for mobile web access. It also included iPod functionality such that it could be used as an MP3 and video player as well. And it of course also had the indistinguishable Apple design. The price for the iPhone started with $499 for the 4GB version and $599 for the 8GB version but was sold a lot cheaper by mobile service providers that also introduced special iPhone data packages. Apple also enhanced the exclusivity of the iPhone by making contracts with single mobile service partners, for example AT&T in the US, giving them a monopoly in exchange for an additional fee per sold iPhone (Siegler, 2010). Although the contract is rumored to last until 2012, Apple agreed to open up in 2011 and also provide iPhones to rival Verizon (Thomson, 2010).

Apple improved the iPhone with each version, getting rid of limitations that were criticized with the earlier versions: The iPhone 3G from 2008 added support for 3G networks (UMTS) and GPS and added an App Store where users could buy, download and install third-party software while developers could easily upload and sell software over that platform. The huge success of the iPhone also made the App Store grow and Apple also gets a percentage of every sold piece of software there. The iPhone 3GS from 2009 improved the hardware with a better processor, more memory and a higher resolution camera and software with the third version of the iOS operating system. The iPhone 4 from 2010 again improved the processor and the display, added a second camera, better battery life and the fourth version of the iOS introduced multitasking[21]. Although iPhones are still a success for Apple and the number of devices is growing, there are still some challenges: RIM is still holding ground, especially for business smartphones (The GigaOM Network, 2010), phones powered by Google's Android are strong competitors (see below), Microsoft has not given up on this segment with the introduction of Windows Phone 7 (see above), Nokia is still working to defend it's position in the market (forming an alliance with Intel (Nokia Blog, 2010) and later with Microsoft (Microsoft, 2011)) and there was also controversy about signal problems with the newest iPhone (Beaumont, 2010).

The Apple TV was first introduced in 2007 as a settop box to connect to the Internet and the local network, streaming audio and video to the TV. The product was upgraded in 2010 with better hardware, better software and the ability to rent movies and tv shows over the Internet (Rigby and Randewich, 2010). Of course it

[21] Multitasking means that many software instances can run "at the same time" either from a system or a user perspective. This concept is common in personal computers since the 1990s but was not used in earlier versions of the iPhone. The reason was seemingly that for multitasking to work well lots of computing power and memory are needed which were lacking in earlier versions of the iPhone.

is not yet clear how the expected convergence of TV and Internet will take place and if Apple can play a role. But it is a development one has to have an eye on, especially as Google is also launching it's Google TV services (see below).

A completely new and revolutionary product is the iPad which was released in 2010. Same think of it as a huge iPhone, some of a new concept for tablet computers and some even as a new concept for notebooks. The iPad is something in between: It also has a multi-touch screen like the iPhone but a lot bigger, being more the size of a tablet or a small notebook. The usability concept is similar to that of an iPhone (using the touchscreen, no keyboard and no mouse) but the larger size makes it more useful for tasks like reading texts (or ebooks), surfing the web and dealing with files.

Although the digital lifestyle products today make up a big fraction of Apple's revenue and profits it is still a innovative computer manufacturer. In 2005 Apple announced that it would switch from the Power PC microprocessors (IBM, Motorola) to Intel microprocessors for it's newest computers starting in 2006 (Apple, 2005). This was an interesting move as Apple was the only major competitor against IBM's Intel-driven platform for over 20 years, leaving the PowerPC processors for gaming consoles, high-end servers and workstations. The reason for the switch according to Apple was the fact that their people liked Intel's roadmap for processors more than IBM's, acknowledging rumors that Apple was not satisfied with the latest Power PC processors. Although Apple still uses it's Mac OS X as the primary operating system in 2006 it announced Boot Camp, a software to install Microsoft operating systems on Apple hardware in parallel (BBC News, 2006).

Looking at some data Apple shows a remarkable growth this decade (Pingdom, 2010). But this is not only due to the diversification of the portfolio but also due to a steady growth in portable and desktop computer segments. Still, quarterly data shows that the new product lines are very important because in the third fiscal quarter of 2010 "iPhone and Related Products and Services“ was making the highest revenues of all of Apple's branches and roughly the same revenue as the totals of desktops, portables and general hardware and software together (Apple, 2010). The total revenue for that quarter is \$15.7 billion and the profit is \$3.25 billion. In May 2010 Apple also overtook Microsoft in market capitalization, making it the biggest tech company in the world at \$222 billion although Microsoft still enjoyed higher revenues and profits (BBC News, 2010b).

4.2.6. Summary, Strategies and Openness

The history of Apple clearly shows the strategy of the company when it comes to it's products: Build sophisticated, sometimes even revolutionary products and sell them for relatively high prices. Market share is just a very secondary goal and the software developed and sold is used to promote the marketability of it's systems (this even includes it's content business - music, tv shows and movies).

This strategy is clearly shown by it's decision (indecision might be the better word) about licensing. Licensing of the superior operating system was not seen as a way of enhancing (long term) market share and profitability but as a problem for the (short term) margins and system sales. Licensing in the eyes of Apple would give competing computer manufacturers the key to catch up and the competition Apple might face would erode profits. Of course, one can argue that building hardware and the operating system together improves integration and therefore product quality and stability but this of course does not rule out licensing to other manufacturers, in fact this ensures Apple a technological lead even when the operating system is licensed because competitors do not have this advantage. The newest Mac OS X versions therefore are still not licensed and cannot be installed on typical IBM-type computers even as due to Apple's move to Intel processors the platform difference is quite small[22].

A source of controversy has been the App Store and it's openness (or lack thereof): Not only does Apple not allow any applications to be distributed to it's

[22] Several hackers and projects showed that with a few tweaks it is possible to run Mac OS X on any Intel machine.

mobile devices from any other platform, it also charges 30% from the price of any software sold over the App Store and has the right to reject any application from the App Store. This includes for example third-party web browsers. Apple also changed the developer guidelines in 2010, making it even tighter: It enforced the use of just a few programming languages and did not allow several scripting languages, effectively locking out alternative development platforms (Northcott, 2010). This is especially interesting because Adobe had already presented a way to translate Flash applications into iPhone applications which would allow development of compatible applications for all major smartphone platforms[23] (Ionescu, 2009). The issue led to an Antitrust investigation by the European Commission (Gamet, 2010). In September 2010 Apple changed the rules again, now allowing those cross-compiling tools in principle such that the European Commission dropped the investigation. This of course does not rule out that cross-compiled applications can be banned in the future considering that Apple can still reject any application it does not like.

Furthermore Apple does not make it's iTunes software compatible to other MP3 players or smartphones which would be possible by using an open interface to which those other products could connect to. This compatibility issue became apparent in 2009 when the Palm Pre smartphone was announced to be able to connect to iTunes by pretending to be an iPhone during the connection. This led to an update of iTunes taking away this possibility but later Palm introduced another update that would again work with iTunes (Paczkowski, 2009). Although it is not clear if this can persist, it was clearly shown that Apple does not want other hardware to be able to use it's platform[24].

Although Apple does not really open up it's systems in terms of compatibility the company has a quite relaxed standing when it comes to open source. Several key products or parts of it's products are open source, for example some components

[23] All other major smartphone platforms can run native Flash applications but the iPhone does not.

[24] This of course makes perfectly sense as Apple makes money with hardware and gives away software and complementary products for a low price or even for free. Allowing other companies access to it's complements can obviously hurt Apple's hardware sales.

of it's operating systems are based on FreeBSD, an open source Unix operating system and the Safari web browser is based on WebKit, an open source browser engine project that Apple is contributing to (Buys, 2010). This blends in very well with the company's overall strategy: For Apple, software is a way to sell hardware and therefore it makes a lot of sense to use open source software as a basis. By using already working open source software it gets a head start in the development and can improve and change it any way it wants without having to pay a fee. This is an even better idea in market segments where Apple is late and has to catch up like in the browser market. Still, Apple of course is not an open source software company: The most distinguished parts of it's operating systems are closed source (like the user interface), lots of key and successful software is completely closed (like iTunes) and I do not know of any open source projects that were started by Apple in the first place. For Apple, open source apparently is a way to improve it's software development, to use it as a basis for some products and to improve from inputs from open source communities.

The last interesting issue about openness is Apple's stance on standards: Clearly, Apple in it's history never was a company promoting open standards, in fact, it often did not care about standards at all (whether open or not) and developed it's own computer technology, it's own operating system, built it's own device connector standard (Firewire instead of USB) and got rid of the floppy disk drives in it's computers when it was still an important feature in other PCs. On the other hand there is a recent initiative that seems to promote not only standards but open standards: Adobe's Flash technology is a proprietary de-facto standard in the web today when it comes to video and user interaction and can of course also be installed on Apple computers. In April 2010 Steve Jobs issued a statement criticizing Flash and promoting the coming open HTML5 standard[25] (Jobs, 2010). Apple's mobile devices like the iPhone and the iPad also do not support Flash but rely on special apps to deliver Flash-like content. Still, when Apple provided a web page showing the capabilities of HTML5 with respect to Flash the demo web

[25] HTML is the standard to write graphical web pages for browsers. It is developed by the W3C and versions of it are released periodically. The standard allows web page creators to develop pages that look (almost) the same on any browser. Although there are differences to what extent browser vendors support the standard it is a huge success.

site would just run on it's own Safari web browser (although many other browsers can already display HTML5) (Otter, 2010). The further developments in this area clearly have to be monitored.

The overall situation in terms of openness is therefore stricter in certain aspects than Microsoft but also more open in others. This is probably due to the different perspective the companies have: Microsoft is a software company and therefore challenged by open source although compatibility or standards can have a positive effect in certain fields, especially when the company is not the monopolist and does not have to defend it's barrier to entry. Apple on the other hand is primarily a hardware company that uses software to promote it's hardware sales: Open source can therefore be a way to improve software while compatibility allows competitors to challenge Apple's premier technology and is therefore often avoided.

4.3. Google

4.3.1. Web Search

The history of Google started in 1995 on the campus at Stanford University where the would-be founders of Google, Larry Page and Sergey Brin met for the first time. Both were graduate students in computer science and they began working on a project they called BackRub in 1996 as part of their graduate program. Google is therefore a lot younger than the other two companies and started in a very different age of computing when the Internet was becoming more and more important. Back then Apple was still in the midst of it's restructuring and Microsoft was trying to find ways into the Internet business.

Back in the day Internet search was not very advanced yet and it was pretty hard to find good information by typing in search queries into one of the then leading search engines like AltaVista or Inktomi. BackRub, based on the PageRank algorithm developed by Page and Brin, was set out to be much better. The breakthrough idea that this was based on was to rank web pages by it's importance and the importance of a web page was based on hyperlinks, connecting one web page to another. If there are many links to a particular web page then this web page is more important than another. And even more, a link from a more important page gives more importance to the linked page than from a less important page. With this (recursive) idea one can rank pages on the Internet and therefore drastically improve the results for a web search. The BackRub project led to a full web search page that was available on the Stanford campus in 1997 and became popular immediately. Later the same year the name was changed to the now familiar "Google", a misspelling of the word "googol" which stands for the number of one followed by one hundred zeros.

Quickly the search engine became so popular at Stanford that the project needed better and more computers to suit the demand for search queries. The technology was under way to be patented and Brin and Page tried to sell licenses for it's superior technology to search engine companies. But none of the firms were interested, not even Altavista which back then was the best and most successful search company and Brin and Page were willing to sell at a modest price. In the end, both were convinced that they had to start their own business to bring their search engine to the market but the funding was a huge problem at first as many did not view search as an important Internet application and it was not clear yet how to make money with web search.

This changed in 1998, just shortly before Google ran out of cash when Andy Berchtolsheim, co-founder of Sun and successful technology investor met with Brin and Page and was so excited about the idea that he handed over a check worth $100,000 to "Google Inc.", a company that hasn't even been created yet. This made Brin and Page decide to take a leave from Stanford to start Google Inc. Google was already very popular although it was still not a finished product and it was not advertised anywhere. In late 1998 it was even included in the Top 100 web sites in the PC Magazine. Still, Google again almost ran out of money in 1999 but due to the great technology Brin and Page were able to get $25 million from venture capital firms Sequoia Capital and Kleiner Perkins and still kept the majority of the company.

The capital allowed Google to grow remarkably in that time (1999-2000), both in terms of personal but especially in terms of computing power. Instead of buying and building up huge super computers they built networks of smaller and cheaper computers, trimming those computers to the minimum to save electricity. Furtermore, they used Linux as the operating system for their servers, avoiding license fees. The whole computer system therefore was very efficient in terms of computing power per dollar which gave Google an advantage. Redundancy was also a very important feature in Google's computer network such that one computer or even one cluster could fail without affecting the operation.

4.3.2. Advertising

Still, with all the new capital, the growth of the customer base and the search index, it was clear that being profitable was a problem for the company. The initial plan was to license the technology to other web-based companies but in 2000 only Red Hat and Netscape did use it for their products. It seemed that nobody was willing to pay for the service although it was very successful already. Although Brin and Page did not have a high opinion on advertising in the search business and there was always the problem of how to avoid the results being skewed to advertising partners they finally decided in 2000 that it was still the best way of making money and they also could keep the search service for free.

The way Google used advertising was still unconventional at that time: It did not use any advertising on the main page (this is still the case today) even though this is the page that is viewed most. It further did not use any banners, graphics or popups but text-only ads. And it separated advertising from search results calling the former "Sponsored Links“ and made sure that the advertising did not interfere with it's free search service. Google called it's program AdWords and offered an automated way for businesses to do advertising at Google. Advertising was even ranked by relevance (just like with the search results) and in that way both the usefulness of the advertising links as well as the advertising revenue were optimized. The system was more like a continuous auction where prices for ads on certain keywords were determined by the willingness to pay of the advertisers. Some keywords were therefore more expensive than others, reflecting the importance of the keyword to Google's advertising partners. However, the company offering most for an ad on a certain keyword did not guarantee the top spot in the "Sponsored Links“ list because the relevance of the ad was always taken into account, too. The advertising system would later be opened to the web, meaning that web masters could sign up at Google and be a partner by including Google's ads in it's web page, sharing the revenue with Google.

The same year Google announced it would provide online search for Yahoo and that by assembling an index of over 1 billion URLs it became the largest search engine in the world. Even more importantly: In a study 99% of the participants identified Google as the best search engine. In 2001 Google expanded the search service to include images and expanded the scope of the business by translating the website in many different languages (72 by 2002) and by putting in a lot of effort to gain advertising customers worldwide. Many worldwide offices opened in the next years, including Tokyo (2001), Sidney (2002) and Dublin (2004). In 2002 Google announced that it would also provide web search for AOL, The New York Times and Amazon, further increasing Google's user base. Around that time "to google“ found a place in English dictionaries meaning to do Internet search. In 2003 Google expanded it's scope even further, introducing Google News, a search engine for news around the web.

Having the products and business plan in place Google started making profit for the first time in 2001: $7 million which increased to $ 100 million in 2002. Given the fact that Google became very profitable and that it was originally funded by venture capital firms and other investors and that many employees had stock in the company, there was a lot of pressure to make Google a publicly traded company. Although the founders seemingly preferred to stay private, the decision was made to go public in 2004. In the first half of 2004 profits were even up to $ 143 million, so there was lots of demand for the initial public offering.

4.3.3. Product Diversity

In that time during the public offering Google became more than a specialized search and advertising company, it became a broad software company. At Google employees were encouraged to use one day a week to work on a separate project of interest. This did not have to be a marketable idea but if it was then Google would make it an official project. The cash flow it received from it's search and advertising business was used to try out many different projectsbut also to buy other start-ups and integrate them into the Google universe. I will present some of the most important products that also made them competitors to many incumbent software and computer companies, especially Microsoft and later also Apple.

In 2004 Google launched GMail, a free web based email service that immediately became a competitor to Microsoft, Yahoo and others. The major innovation was to offer huge amounts of webspace for those emails, starting with 1GB of storage which at the time was 500 times the amount Microsoft offered. For Google the service allowed them to better utilize it's huge computer network and earn even more with advertising. But the advertising was a problem because to do advertising with emails one has to process those emails and check which type of ads might be profitable to display to the customer. This seemingly was the first time Google got into problems with data privacy when commentators and customers were unsure about how this processing was to take place, if emails were stored and cataloged long-time, if they were even read by people at Google or the government. Google always denied such claims and GMail still became a popular email service but some uncertainty was created about Google's use of data. I will touch that issue in the next subsection.

In 2002 Google started discussing a project called Google Print (later renamed to Google Book Search). The goal was to index books present in libraries and make them searchable in a very general way. The first version of the service was launched in December 2003 and a year later Google announced the partnership with the University of Michigan, Harvard, Stanford, Oxford and the New York Public Library. Although scanning all those books was tricky from a technical standpoint, the probably more challenging issue was copyrights. In 2006 Google made works downloadable that were out of copyright but by making copyright protected books searchable and publishing parts of those works Google moved itself into some grey area in terms of copyright protection law. This was even more of an issue as Google was making advertising money with those searches. This led to infringement suits by both the American Association of Publishers (AAP) and the Authors Guild (AG) in 2005. A settlement was reached in 2008, allowing Google to scan, index, search and display portions of books and make out-of-print books available for download even if those were still copyright protected (Reed, 2009a). Google was also paying $15,500,000 to the AAP and committed to other payments related to the digitalization of the books before the settlement was reached and further agreed to share advertising revenue with authors and publishers[26]. However, there is also criticism about that settlement, especially as it has the potential to give Google a monopolistic advantage because potential competitors are not subject to the same rights that are granted to Google (Reed, 2009b). Some of those issues are still to be resolved.

The first important desktop software product by Google was a toolbar for web browsers that was released in 2001 to allow easier access to Google's web search. In 2004 however Google challenged Microsoft by releasing Desktop Search, a software that uses Google search technology to index and search files stored on the computer harddisk. Although Microsoft improved it's desktop search technology in later Windows products the free to download Desktop Search is still available.

Later in 2004 Google acquired Keyhole, renamed it to Google Earth and eventually released the basic version for free. The software lets users search for addresses and locations on earth and shows 2D or 3D satellite and aerial pictures of that

[26] More detailed information as well as original texts of the settlement and further procedures can be found at www.googlebooksettlement.com.

location, including historical images and other features. Just a few months later Google released Google Maps, an online service that includes map searches and features like a route planner or satellite images. It also allows the search for other locations around that area, for example for hotels, restaurants or doctors. In 2007 Google added to that by announcing Google Street View which allows users to explore neighborhoods by watching pictures made from street, usually by a car.

In 2005 Google acquired Urchin, a company that developed a web analytics software that then became Google Analytics which allows web masters to optimize their website by providing them information about their users or customers. With this tool web masters can improve the marketing of their web sites, optimizing their revenue with advertising which of course also benefits Google.

In 2006 Google acquired YouTube, the most popular video sharing community on the Internet and paid $1.65 billion in stock. YouTube is still using it's own brand name although the community now uses Google accounts to sign up and Google's advertising system. Although very popular it was not clear whether YouTube was making any profits before and after the Google acquisition but it was assumed that this was not the case until at least 2010 when analysts predicted a positive profit (Kafka, 2010). However, real data from Google does not seem to be available. As with Google Book Search the service is problematic in terms of copyright: Users can upload any video they want. Even though they are informed that they are not allowed to upload videos without permission of the copyright holders, still, there seem to be many videos that are not permitted under copyright law. Owners of those rights can request the deletion of the video when they find out about the copyright infringement though. From the point of view of Google it is not the service that is possibly breaking the law but users who use that service. Still, many firms sued Google and YouTube, the most notable being Viacom in 2007 over $ 1 billion in damages. In 2010 however, a US judge ruled that the service was protected by provisions of the Digital Millennium Copyright Act (Lefkow, 2010). Viacom decided to appeal however and the case is still ongoing.

Also in 2006 Google bought Upstartle, maker of Writely, a web-based software to write text similar to the desktop-based Microsoft Word. This became the basis of Google Docs. Later that year Google introduced Google Docs and Spreadsheet, combining Google Docs with a web-based spreadsheed application ala Microsoft Excel. The presentation software Presentely was added in 2007 and a drawing and a forms application would follow. Today, Google can offer a web-based portfolio of free productivity applications, making it a competitor to Microsoft Office. The use of data is again questioned as all data is stored on Google servers. Of course this makes files accessible from anywhere and easier to collaborate when people are working on the same file, also. Google Calender is another web-based application that allows users to store calender entries and tasks which was introduced in 2006 as well.

In 2007 Google launched the Open Handset Alliance and announced Android, an open source operating system for mobile devices. Joining Google in that alliance were 30 firms, Motorola, Qualcomm, HTC and T-Mobile among them. Android is based on Linux and includes the basic operating system, a user interface and some applications like phone or text software. The Open Handset Alliance also offers a Software Development Kit (SDK) for programmers and software companies to write software (Apps) for the Android operating system. Furthermore, Google offers an Android market where developers can store their software and sell it to Android users via Google (Google, n.d.b). Google has the same 30% transaction fee of the price of the software like Apple (Chu, 2008). However, Android is much more open as one can also install software without using the Android market. The first phone using Android and built by HTC was announced by T-Mobile in September 2008 and many other phone making companies like Samsung, Motorola, LG, Sony Ericsson and others followed. The platform took off which also showed in the number of available Apps which reached about 200,000 in January 2011 (Android Market, n.d.). Since summer 2010 more Android phones are sold than phones from Apple (Tofel, 2010).

In January 2008 Google also launched it's own web browser called Chrome, again a free and open source product. Chrome is built on some other open source browsers like WebKit and Firefox but also introduced some new features, for example a very fast JavaScript engine. Chrome is available on all major operating system platforms and became a favorite to some users due to it's speed and security features. Having it's own web browser gives Google more independence because it can then shape the future of web browsing which is of course important as all of it's business interests are on the Internet. Data about browser usage shows that Chrome has become the third-most popular web browser (behind Internet Explorer and Firefox) and is constantly increasing it's usage share[27].

Google also acquired On2 Technologies in February 2010, creator of a high-quality video compression software called VP8. Together with the already open audio codec Vorbis and the open container format WebM the WebM project financed by Google wants to build a free and open media standard to be used on the web, in contrast to the closed MPEG standards. Chrome, Firefox and Opera exclusively use WebM for the new HTML 5 standard while Internet Explorer and Apple Safari use MPEG standards as their first choice although they can also use WebM (Blizzard, 2010; Mills, 2010; Hachamovitch, 2010; Bankoski, 2010; Gasston, 2010). Of course it helps that Google is controlling the web's most important video site YouTube which also offers WebM videos (YouTube, n.d.). Whether WebM can become the web video standard of the future remains to be seen. A problem is that it is still not clear whether VP8 violates patents held by other companies and if it can offer the same quality as the new MPEG standards (Daffara, n.d.).

Based on it's success with the Chrome web browser Google announced the Chrome OS project, renamed in 2009 to Chromium OS (although the product is still called Google Chrome OS). The goal of this project is to build an open source Linux-based operating system that is very fast and secure and allows easy Internet access. It is not the goal to install software on it like on a conventional operating system but applications are used via a web browser and are accessed on the web. Although the operating system is still not used very much there is of course the possibility that it might compete with Microsoft Windows in the future, especially if more applications or computer usage in general shifts towards the Internet.

[27] Data sources and discussion can be found in the section about Microsoft and the Internet Explorer

The latest product that has high potential is Google TV, announced in May 2010, a system that is built on Android and Chrome and wants to marry TV and the web, allowing TV users to also watch web channels but also regular web sites and social networks. Again, the system is open and Google is working with industrial partners to get Google TV onto settop boxes and TVs and with software developers to improve the user experience. Although it is still very early and the success of this approach cannot be predicted as of now it adds to Google's open projects with potential high reward in the future.

But Google also did not stay put in it's most important products, improving the search technology, enlarging the index of web sites and improving the design of it's web site several times. In web advertising, Google became even stronger after acquiring DoubleClick, it's strongest competitor and leading company in "display ads“ in November 2008[28]. In June 2009 Google announced AdSense for Mobile, expanding it's advertising program to mobile platforms and allowing developers to display advertising in their mobile applications. In May 2010 Google also acquired AdMob, a leading company in mobile advertising to further strengthen it's business.

The fast pace that Google is working on projects of course also leads to failed projects, most of them of course never leave the Googleplex or any other lab that Google researchers and engineers are working in. One that did and still failed was Google Wave, a service launched as a preview in May 2009 with the very optimistic idea to revolutionize email. The service is a combination of email, collaborative text creation, chat and social networking. However, the project was shut down in April 2010 as Google did not see the user reception it wanted. Still, Google announced that it wants to use parts of the product within other Google projects and released the code as open source.

[28] Google announced the acquisition a year earlier but regulators stopped that process. Although in the end the merger was allowed it clearly showed that there were concerns about dominance in the market for web based advertising.

4.3.4. Summary, Strategies and Openness

Google's growth from a small start-up out of the University of Stanford to one of the biggest software companies in the world has been remarkable. Based on it's dominance in search technology and web advertising Google has diversified it's product base in the last several years, moving into competition with the biggest computer companies. Google's lead in search is not primarely based on any consumer lock-in but on superior technology and it's lead in advertising is based on the ease of use and huge network size: Given the large amounts of visits to Google and it's partner sites already very small numbers of advertising clicks and click fees can generate a strong revenue.

Google's products are based on open source, both as a basis for it's services like the Linux operating system and as a licensing choice for many of it's products. Furthermore, many of Google's products and services use open standards or (if there are none) define some. However, the most important revenue generators for Google, it's search technology and advertising system, are not open. In a blog post Jonathan Rosenberg, Senior Vice President of Product Management at Google, explains some of Google's views on openness and how it relates to it's strategies (Rosenberg, 2009). Although arguing that "open systems win“ and that with new products Google should use or define open standards and open up it's source code and interfaces Rosenberg argues that this should not hold for the two mentioned products/services. The reasons are that the markets for search and advertising are already very competitive and therefore opening up would not increase competition. Furthermore, if the ranking algorithms were open people could manipulate search results or advertising lists. Although I accept the second argument as valid I am not sure whether so much would change because search engine optimization is already a marketing tool on the web, even when the algorithms aren't open. I would rather assume that opening up those technologies for which Google is almost a monopolist would allow other companies to copy it's technologies and close the technology gap or even just offer the same technology with a different design or business plan. Clearly, Google would have a lot to lose if due to higher competition the traffic on Google's search engine becomes lower or partner sites work with different advertising providers.

The main argument by Rosenberg for opening up all those other products is scale. By opening up and basically allowing other companies to use it's technology Rosenberg argues that this leads to a lot of innovation which eventually increases the business for everyone: "We may get a smaller piece, but it will come from a bigger pie“. I am not sure whether this holds true in general, taking into account that the products are for free. For Google however, this really could hold true at least when the new technology leads to higher usage and more Internet time by the costumers. This in turn leads to higher overall traffic and quite a fraction of it will land either at Google or on partner sites that use Google's advertising service, generating revenue for Google.

A different issue is openness in terms of data: Google is a company that depends on information and therefore also has to store lots of information. I touched on the issue that appeared when Google started using information stored in emails from the GMail service to display advertising. Google uses a single membership account for all of it's services which is convenient but also raises privacy concerns. People that use GMail, store texts and pictures and offer videos on Youtube could be profiled if that data is combined. Google therefore started issuing it's privacy principles in January 2010 and is updating those principles on a regular basis, claiming that the data used by Google is secure and that it will also not be shared with other companies. Furthermore, Google Dashboard lets users control the data that is stored by Google. Another problem appears due to Google's services that are used on other web sites, mostly advertising and Google Analytics, that could potentially be used to track user's surfing habits. And then, search results could in principle also be skewed to favor certain web sites.

The overall picture for Google therefore looks very clear: Google is an open company and it is also very open in saying that it is profitable for them. Interestingly enough that the most important parts of it's business are not open even though they are for free (web search) or open in terms of letting partner sites participate (advertising). The story for Google's success is therefore a story about external revenue, a perspective that I will discuss in more detail below. Whether Google can stay that open forever or if this is just a temporary strategy as long as the advertising business is working well is of course yet to be seen.

4.4. Summary

Reviewing the history of the three mentioned companies shows that the strategies in terms of openness are not only different but they also changed over time. Microsoft's early success is based on portability and IBM's open platform but then it built up a software barrier to entry to monopolize the operating system market and used it's closed operating system monopoly to take over other markets as well offering closed products, interfaces and standards. Although this strategy proved successful the legal problems might force Microsoft to change it's approach to openness and some signs already point in a more open future (see above). That some of it's most important products might not be as important in the future due to technological change might also force Microsoft to further evaluate it's strategy.

Apple on the other hand has always been a closed company due to the fact that it sees itself primarely as a hardware company. For Apple software and services are ways to improve hardware sales such that a closed approach (no compatibility with other products, different interfaces and standards) seems optimal. That Apple nevertheless sometimes uses and works with open source software can also be understood in this way because it allows Apple to improve it's offers by giving the company a head start in certain developments (operating systems, web browsers). However, a necessary condition for this approach seems to be a lead in technology (e.g. quality of products) because otherwise customers can choose the product with the better compatibility (see above for the crisis in the 1990s).

Google on the other hand has always been an open company, driven by the external revenue generated by it's advertising business. The major exception to this strategy is it's search algorithm which is not an open technology although other companies can license it. The strategy makes sense for Google as due to it's strength in advertising revenues grow with the overall Internet usage and not with Google's market share in any particular market. It will be interesting to monitor whether this approach is sustainable in the future.

5. Economic Analysis

The economic analysis of some of the issues and topics presented in the last section will be done by using a game theoretic model with several stages. Taking into account suggestions from related work I develop a duopoly model that combines vertical product differentiation with an incumbent-entrant-scheme and featuring some network economics based on and inspired by the model by Baake and Boom (2001). Although I do not fully solve the model for all possible parameter values due to it's complexity, I show that I can discuss a wide range of topics related to openness by altering certain details and parameters of the model.

5.1. The Model

5.1.1. General Methodology

In this model there are two firms ($i = 1, 2$) that compete in a market for consumers that have a certain valuation for the products (I also use the word systems) of those firms. Firm 1 is the incumbent that has already been able to sell N_1 of it's systems in the past[29]. Firm 2 has developed a different system that is not compatible to the first one and can enter the market to compete with the incumbent. Furthermore, given the topic to discuss, the potential entrant also has other options which I will discuss later. Note that the development of the second system is exogenous as well so I do not explain how firm 2 got the innovation, one can assume it to come from a research or development project. Firms then set prices competitively and offer the products to the consumers. In the last stage consumers decide whether to buy one of the two systems or no system at all. The result of the model is a subgame-perfect equilibrium meaning that the decisions are optimal in all stages for all firms and consumers given the decisions of the other consumers and the other firm. Note

[29] Of course, one could also endogenize how the incumbent sold those systems in an additional stage but I believe this does not add much insight to the model.

that I am only discussing pure-strategy equilibria. Firms are able to assess what will happen in following stages and base their decisions on that information. The model can therefore be solved using backwards induction and an equilibrium is subgame-perfect if in every stage an equilibrium is played. In this type of model for general settings of parameter values not only different equilibria can occur but also non-unique equilibria or no equilibrium at all. I will now formalize the properties of the different stages of the model.

5.1.2. Stage 1: Licensing Decision

In the first step, firm 2 has to decide what to do with the technology, given that firm 1 is the current monopolist. Firm 2 can use the technology for it's own purposes which is the standard case. In certain topics firm 2 can also offer a license to the current monopolist or release it as open source.

If the technology is used by the company itself, it can enter the market in the next stage and compete against firm 1. Prices, revenues and profits of the now duopoly market are determined in later stages. Licensing gives the firm a profit of $\Pi_2 = T$, a fixed license fee that it gets from firm 1. Firm 1 then stays monopolist and earns the monopoly profit minus the license fee. I assume that firm 1 will always agree to the licensing offer if the fee is lower or equal to the (positive) difference in profits for firm 1. I assume that there are no other licensing options (e.g. volume based licensing) and by selling the license firm 2 grants an exclusive right to it's technology to firm 1 so that it cannot produce on it's own.

Releasing the system as open source means that the technology is opened to the public and everybody can use it for free. This can be optimal for firm 2 if either it cannot make a profit by entering or if the goal is not to maximise the profit or if the revenue comes from an external source. I will elaborate more on those situations below.

Depending on the situation I assume different cost structures:

1. As described in the very first section systems in information technology tend to have large fixed costs and relatively small variable costs and this holds true even stronger for software systems than for hardware. I therefore normalize the variable costs to zero and this is equal for all firms that produce systems.

2. Furthermore, I assume that the incumbent has already paid any fixed cost before entering the market (before this game has started).

3. However, I assume that there are other positive fixed costs, depending on the situation: F_1 are the fixed costs that firm 1 has to pay if it takes over the new technology via a licensing agreement. Most of those costs can be interpreted as costs to make the technology compatible to their own technology such that the existing network can be preserved[30]. F_2 are the fixed costs for firm 2 if it enters the market in stage 2. Those can be costs for advertising, documentation, support or even a production facility (for example for installation discs).

4. If firm 2 just releases it's technology as open source for example on the Internet without actually marketing or supporting it's product, then the costs probably go to or are close to zero. Therefore, I assume that an open source release is costless.

5.1.3. Stage 2: Entry Decision

If firm 2 chooses not to license and not to open source it's technology then it can use the technology to build a product and enter the market. It has to decide between 2 options here, taking into account what will happen in the next stages of the game.

[30] The term "network" be will defined later.

1. Stay out of the market: $\Pi_2 = 0$

2. Enter the market and compete against the other firm: $\Pi_2 = p_2 z_2 - F_2$

p_2 is the price set by firm 2 in stage 2 while z_2 is the number of systems sold (or number of consumers who bought that system) in stage 3. I will also call z_2 the "network“ of firm 2 while N_1 is the existing network and z_1 is the additional network for firm 1. Staying out of the market in this stage is of course only optimal if the fixed costs are so high that they cannot be earned by the revenue that is earned in the later stages. If there was a licensing agreement in stage 1 or firm 2 opted to release it's product as open source then this stage is skipped.

5.1.4. Stage 3: Pricing Decision

If firm 2 enters, there is a duopoly market and both firms set prices competitively. An equilibrium is reached if the prices are mutually optimal given the price of the other firm. The profit for firm 1 is $\Pi_1 = \pi_1 = p_1 z_1$ in this case with p_1 the equilibrium price of firm and z_1 the network of firm 1.

If firm 2 does not enter and also did not license the system or open source it, then firm 1 is a monopolist in the old technology with the same profit function as above. If firm 1 licenses the technology by firm 2 it is also monopolist but the profit is then $\Pi_1 = \pi_1 - F_1 - T$, the revenue minus the fixed costs that come from using the new technology and minus the licensing fee that goes to firm 2.

In the open source case firm 1 has to compete with a product that is offered at zero price: $p_2 = 0$. The profit is again $\Pi_1 = \pi_1 = p_1 z_1$ although the price and network most likely will be different than in the competitive case. The profits of both firms in the different cases will be discussed in more detail below.

Note on the notation: Π_i is the overall profit by firm i that also includes all fixed costs while π_i is the revenue for firm i (the profit if the game would only consist of stages 3 and 4).

5.1.5. Stage 4: Consumer Choice

Consumers are differentiated in the way they view the systems: All consumers value one system over another but have a different willingness to pay. I define utility functions for up to two system ($i = 1, 2$):

$$U_i = s_i x + \beta(N_i + z_i) - p_i$$

and $U = 0$ if no system is bought. s_i is the quality of system i, N_i the number of systems i that have been sold already or the already installed network for that system[31], z_i the number of systems i that are sold in this stage and p_i the price of system i. Users are differentiated in terms of their willingness to pay for quality and are independently and identically distributed: $x \sim [0, 1]$. Consumers buy the system that gives them the most utility or no system at all if the utility for both systems is negative.

One can clearly see the different features of the model in these utility functions: Vertical differentiation among products, the network economics part and an incumbent-entrant structure. An equilibrium in this stage is defined by a set of networks (z_1, z_2) such that no single consumer has an incentive to switch it's system choice by changing the network or dropping out of it. The network economics nature of those utility functions allows for many different equilibria and multiple equilibria can also occur given the choice of model parameters.

[31] I always assume $N_2 = 0$ so that if firm 2 is in the market it has no existing network.

5.1.6. Differences to the Original Model

I just want to quickly summerize the major differences to the model by Baake and Boom:

1. I do not assume that both firms come into the market at the same time but that one firm is already in the market and has already built up a network.

2. The other model endogenizes the quality decision of the firms while I assume that this is done exogenously before the game starts. Each firm has a technology/development at hand and can use it in a way the model allows it.

3. In the other model there are marginal costs that are proportional to the quality of the product and no fixed costs. I assume some fixed costs but neglect marginal costs.

4. The consumer model does not assume increasing network returns to system quality but instead the network effect is independent of the "quality effect" and vice versa. Baake and Boom do not give any explanation or reason for their assumption such that it makes sense in my opinion to leave the two effects independent of each other in the utility functions for the consumers.

5. Baake and Boom discuss compatibility and quality decisions by the firms while in this model more general decisions about openness can be discussed.

5.1.7. Parameter and Decision Choices

The model allows for many different parameter combinations that lead to different equilibria in different stages. I therefore restrict the parameter space a little by using the following assumptions:

I assume that the technology by the potential entrant is better than the one by the incumbent and the other technology is normalized to a quality of 1: $s_1 = 1$ and $s_2 = s > 1$. In terms of the network effect I further assume $\frac{1}{2} < \beta < 1$ such that the marginal utility from the network is lower than the utility of the system quality. Furthermore, I restrict $s > \beta + \frac{1}{1-\beta}$ such that the quality difference is relatively large. Note that the restriction enforces $s > 1$ automatically. The assumptions lead to unique equilibria in stage 4 but also facilitate the process of finding equilibria in stage 3 which will be shown below.

5.2. Stage 4: Consumer Decision

As the model will be solved using backwards induction I start with the analysis of the last stage. The topics I discuss later on using the full model have the same properties with respect to stage 4 and therefore the solutions here can be used as general results for all topics. The decisions in the earlier stages are already fixed in this stage, meaning that the licensing and entry decisions were already made and that prices were already picked in stage 3. Therefore, in stage 4 there are either 1 or 2 systems in the market and the prices of those products are fixed. Consumers decide on which system to buy (if any) given those prices and given the choices of the other consumers.

5.2.1. Two Systems

If there are two systems for the consumers to choose from with prices (p_1, p_2) then there are 7 different equilibria that can be played by the consumers. I number the equilibria from 1a, 1b to 4 with the following notation: An "a“ equilibrium is an equilibrium where there are consumers who do not buy any system while for "b“ equilibria the full market is served. The number in the beginning denotes the sharing of the market (same notation as Baake and Boom): In equilibria 1a and 1b both firms sell system while in equilibrium 2a/2b only firm 1 sells and in 3a/3b only firm 2. In equilibrium 4 consumers do not buy any system. Equilibria are defined by the number of sold systems (z_1, z_2) such that $z_i \geq 0$ and $z_1 + z_2 \leq 1$ holds. The different equilibria are summarized in Table 1 together with the restriction on prices (if applicable). Restrictions on prices means that certain equilibria only occur in certain ranges of (p_1, p_2) and furthermore certain inequalities have to hold. I will now formally show the properties and conditions for the different equilibria.

In equilibrium 1a $(z_1 > 0, z_2 > 0, z_1 + z_2 < 1)$ both systems are sold: System 1 is sold to the consumers with low quality valuation and system 2 to the ones with high quality valuation and there are also consumers who do not buy any system. Due to the nature of the utility functions the space of the consumer valuation is partitioned by two distinct points x_1, x_2 with $z_1 = x_2 - x_1$ and $z_2 = 1 - x_2$ that are defined as follows:

$$U_1(x_1) = 0 : x_1 + \beta N_1 + \beta z_1 - p_1 = 0$$
$$U_1(x_2) = U_2(x_2) : x_2 + \beta N_1 + \beta z_1 - p_1 = s x_2 + \beta z_2 - p_2$$

This is a set of linear equations for x_1 and x_2 given (p_1, p_2). If now $0 < x_1 < x_2 < 1$ then this constitutes an equilibrium as every consumer does best given the choice of the other consumers. All consumers with $x_2 \leq x \leq 1$ buy system 2, consumers with $x_1 \leq x < x_2$ buy system 1 and consumers with $x < x_1$ don't buy at all. I assume that indifferent consumers (either $x = x_1$ or $x = x_2$) buy the system with the higher quality. The restrictions on x_1 and x_2 lead to three restrictions on (p_1, p_2) and combining these restrictions also enforces a necessary condition for the range of both prices where this equilibrium can occur (see Table 1).

Equilibrium 1b $(z_1 > 0, z_2 > 0, z_1 + z_2 = 1)$ is similar to 1a but the full market is served. Therefore, for $x = 0$ the utility function for system 1 has to be nonnegative. $z_1 = x_2$ and $z_2 = 1 - x_2$ and $0 < x_2 < 1$ has to hold for this equilibrium. x_2 and the price restrictions are determined by:

$$U_1(0) \geq 0 : \beta N_1 + \beta z_1 - p_1 \geq 0$$
$$U_1(x_2) = U_2(x_2) : x_2 + \beta N_1 + \beta z_1 - p_1 = s x_2 + \beta z_2 - p_2$$

This is an equilibrium if $0 < x_2 < 1$ and then all consumers with $0 \leq x < x_2$ buy system 1 and those with $x_2 \leq x \leq 1$ buy system 2.

In equilibrium 2a ($0 < z_1 < 1, z_2 = 0$) only system 1 is sold and no consumer buys system 2. For this to happen, the utility function for system 1 has to always be larger than the one for system 2, such that even the consumer with the highest valuation of quality ($x = 1$) buys system 1. Then, $z_1 = 1 - x_1$ for $0 < x_1 < 1$ and:

$$U_1(x_1) = 0 : x_1 + \beta N_1 + \beta z_1 - p_1 = 0$$
$$U_1(1) \geq U_2(1) : 1 + \beta N_1 + \beta z_1 - p_1 \geq s - p_2$$

If $0 < x1 < 1$ holds then this is an equilibrium and all consumers with $x_1 \leq x \leq 1$ buy system 1.

In the 2b equilibrium ($z_1 = 1, z_2 = 0$) again only the first system is sold but now all consumers buy that system. This happens if the consumers with the lowest valuation of quality ($x = 0$) also has a non-negative utility:

$$U_1(0) \geq 0 : \beta N_1 + \beta - p_1 \geq 0$$
$$U_1(1) \geq U_2(1) : 1 + \beta N_1 + \beta - p_1 \geq s - p_2$$

The first inequality already defines the price range for firm 1 while the second defines a price restriction.

In equilibrium 3a ($z_1 = 0, 0 < z_2 < 1$) only the second system is sold while no one buys system 1 and some consumers don't buy at all. For this to happen, the utility function for system 2 has to be always larger than the one for system 1 whenever the utility function for system 2 is nonnegative. Then, $z_2 = 1 - x_2$ for $0 < x_2 < 1$:

$$U_2(x_2) = 0 : sx_2 + \beta z_2 - p_2 = 0$$
$$U_1(x_2) \leq 0 : x_2 + \beta N_1 - p_1 \leq 0$$

If $0 < x_2 < 1$ holds then this is an equilibrium and all consumers with $x_2 \leq x \leq 1$ buy system 2.

Equilibrium 3b ($z_1 = 0, z_2 = 1$) is similar but the full market is served:

$$U_2(0) \geq 0 : \beta - p_2 \geq 0$$
$$U_1(0) \leq U_2(0) : \beta N_1 - p_1 \leq \beta - p_2$$

Equilibrium 4 ($z_1 = 0, z_2 = 0$) is the one where no consumer buys anything. This happens if the utility functions are smaller than or equal to 0 in the full range of x, or equivalent, for the consumer with the highest valuation of quality ($x = 1$):

$$U_1(1) \leq 0 : 1 + \beta N_1 - p_1 \leq 0$$
$$U_2(1) \leq 0 : s - p_2 \leq 0$$

In this case there is no price restriction but the inequalities already show the price ranges (again, see Table 1).

Proposition 1. *For any price combination (p_1, p_2) an equilibrium in the consumer stage exists and this equilibrium is always unique.*

Proof. If both $p_1 \geq \beta N_1 + 1$ and $p_2 \geq s$ then an equilibrium always exists as then the consumers play equilibrium 4. If one of the two inequalities is not fulfilled then one of the other equilibria is played: If $p_2 > s - 1$ then either 1a, 2a or 2b are played depending on p_1. As the three price restrictions make up the full range of possible values for p_1 (which can easily be checked by inspecting Table 1) an equilibrium always has to exist. For $\beta < p_2 \leq s - 1$ 1a, 1b, 2b or 3a are played and again, the price restrictions make up the full range of values for p_1. The same is true for $p_2 \leq \beta$ when 1b, 2b or 3b are played. Therefore, for every p_2 any p_1 always leads to an equilibrium which proves that an equilibrium always exists for any price combination.

Equ	z_1	z_2	Restrictions on $((p_1,p_2))$	Price Range
1a	$\frac{p_2-(s-\beta)p_1+\beta(s-\beta)N_1-\beta}{(s-\beta)(1-\beta)-1}$	$\frac{p_1-(1-\beta)p_2-\beta N_1-1+s(1-\beta)}{(s-\beta)(1-\beta)-1}$	$p_2 > (s-\beta)p_1-\beta(s-\beta)N_1+\beta$ $p_2 < \frac{s-\beta-1}{\beta}p_1-(s-\beta-1)N_1+\beta$ $p_2 < \frac{1}{1-\beta}p_1-\frac{\beta}{1-\beta}N_1+s-\frac{1}{1-\beta}$	$\beta N_1 < p_1 < \beta N_1+1$ $\beta < p_2 < s$
1b	$\frac{p_2-p_1+\beta N_1-\beta}{s-2\beta-1}$	$\frac{p_1-p_2-\beta N_1+s-\beta-1}{s-2\beta-1}$	$p_2 \geq \frac{s-\beta-1}{\beta}p_1-(s-\beta-1)N_1+\beta$ $p_2 > p_1-\beta N_1+\beta$ $p_2 < p_1-\beta N_1+s-\beta-1$	$0 \leq p_1 < \beta N_1+\beta$ $0 \leq p_2 < s-1$
2a	$\frac{-p_1+\beta N_1+1}{1-\beta}$	0	$p_2 \geq \frac{1}{1-\beta}p_1-\frac{\beta}{1-\beta}N_1+s-\frac{1}{1-\beta}$	$\beta N_1+\beta < p_1 < \beta N_1+1$ $p_2 > s-1$
2b	1	0	$p_2 \geq p_1-\beta N_1+s-\beta-1$	$0 \leq p_1 \leq \beta N_1+\beta$ $p_2 \geq -\beta N_1+s-\beta-1$
3a	0	$\frac{-p_2+s}{s-\beta}$	$p_2 \leq (s-\beta)p_1-\beta(s-\beta)N_1+\beta$	$p_1 > \beta N_1$ $\beta < p_2 < s$
3b	0	1	$p_2 \leq p_1-\beta N_1+\beta$	$p_1 \geq \beta N_1-\beta$ $0 \leq p_2 \leq \beta$
4	0	0		$p_1 \geq \beta N_1+1$ $p_2 \geq s$

Tabelle 1: Possible Equilibria in the Consumer Stage and Price Restrictions

The proof for the uniqueness of the equilibria is shown in the appendix. The crucial assumption for the proof is that $s > \beta + \frac{1}{1-\beta}$ meaning that the difference between qualities has to be strong enough. This qualitatively replicates the result by Baake and Boom in that stage. □

5.2.2. One System

When only incumbent firm 1 is in the market selling to consumers then consumers only have to choice to decide between buying that one system or not buying at all. The quality of the incumbent's system is a general t which could be the original technology of the incumbent ($t = 1$) but also the technology of firm 2 if licensed ($t = s$). The situation is less complex as there are only three equilibria in this case.

In equilibrium 1 ($0 < z_1 < 1$) some consumers buy the system while others don't. Given the nature of the utility function there has to be an x_1 with $0 < x_1 < 1$ and then $z_1 = 1 - x_1$:

$$U_1(x_1) = 0 : tx_1 + \beta N_1 + \beta z_1 - p_1 = 0$$

This leads to $x_1 = \frac{1}{t-\beta}p_1 - \frac{\beta}{t-\beta}N_1 - \frac{\beta}{t-\beta}$ and the restrictions and price range are given by $0 < x_1 < 1$ to $\beta N_1 + \beta < p_1 < \beta N_1 + t$. $z_1 = 1 - x_1 = \frac{-p_1+\beta N_1+t}{t-\beta}$.

In equilibrium 2 ($z_1 = 1$) the full market is served which means that even the consumer with the lowest valuation of quality has a non-negative utility:

$$U_1(0) \geq 0 : \beta N_1 + \beta - p_1 \geq 0$$

This leads to $p_1 \leq \beta N_1 + \beta$.

In the case of equilibrium 3 ($z_1 = 0$) no consumer buys which means that even the consumer with the highest valuation has a non-positive utility:

$$U_1(1) \leq 0 : t + \beta N_1 - p_1 \leq 0$$

The price range is therefore $p_1 \geq \beta N_1 + t$.

Proposition 2. *For any price p_1 an equilibrium in the consumer stage exists and it is also unique.*

Proof. By inspection of the price ranges it is already clear that the equilibria are unique for any values of t, β and N_1. Furthermore, any price $p_1 \geq 0$ leads to an equilibrium as the unification of the price ranges makes up the full range. □

5.3. Operating System Licensing

I now turn to solving the full model for certain cases of interest. One interesting issue that can be analyzed with this model is the licensing of systems/technologies in information technology. One story that comes to mind is the competition in the 1980s when Microsoft had the dominant operating system that was sold on all IBM-type PCs at that time when Apple developed the MacIntosh with a new operating system, including a graphical user interface and many other features that were new for personal computers[32]. The new operating system was so advanced that Microsoft needed until the early to mid 1990s (Windows 3.1 or Windows 95) to include the same features and even asked Apple to license their operating system to other vendors, realizing that it was way better than DOS. Still, Apple did not do it because they saw itself as a hardware company and believed that the better operating system should boost Apple hardware sales.

I analyse this situation using the model at hand. Here, Microsoft is the incumbent (firm 1) and has an already set up network size N_1. Apple is firm 2 and has the

[32] There were earlier developments similar to the MacIntosh but they were sold at a much higher price. The details of that story can be read above in the section about Apple's history.

better product quality ($s_2 = s > \beta + \frac{1}{1-\beta} > 1 = s_1$). The network part of the model is the software developed for the respective system much like the standard approach taken by related papers (for example Church and Gandal (1992)). I use the analysis of the fourth stage from above and solve the rest of the model using those results.

In stage 1 firm 2 has to decide whether to offer a licensing contract to firm 1. If also firm 1 agrees then firm 1 is the monopolist with the new techology with quality s and has to pay fixed costs $F_1 > 0$ to make the system compatible to it's existing network N_1. When $T > 0$ is the licensing fee then

$$\pi^L - \pi_1 \geq T + F_1$$
$$max\{\pi_2 - F_2, 0\} \leq T$$

has to hold such that licensing is optimal for both firms. π^L denotes the monopoly revenue for a monopolist by selling quality s when it licenses firm 2's technology and π_i is the revenue for firm i when there was no licensing agreement. The conditions mean that both firms have to be better off with this licensing agreement which depends on whether firm 2 would enter in stage 2 or not.

If there was no licensing agreement then firm 2 has to decide whether to enter the market and sell quality s or stay out. The fixed costs $F_2 > 0$ to enter have to be paid in this stage. Clearly, entering is only optimal when the revenue is at least equal to those fixed costs. Those revenues therefore decide whether firm 2 can enter the market with it's own technology and whether licensing is profitable.

5.3.1. Stage 3

Given the different options in stages 1 and 2 there are three possibilities: Firm 1 could be a monopolist with the worse (no licensing and no market entry) or better (licensing) technology or there could be a duopoly market with both firms selling their own technology. Firms maximize their profits given the price of the other firm (in the duopoly case) and the equilibria that are played in stage 4 by the consumers. A vector (p_1, p_2) is an equilibrium in this stage (subgame) if the pricing decisions are mutually optimal for both firms, meaning that p_1 optimizes the profit of firm 1 given p_2 and vice versa. If only the incumbent is active in the market, then an equilibrium is defined by a price p_1 that maximizes firm 1's profit. Note that as costs are fixed optimizing the profit is equivalent to optimizing the revenue defined by $\pi_i = p_i z_i$ $(i = 1, 2)$.

Proposition 3. *If the incumbent is a monopolist in this stage selling quality s to the consumers (the licensed technology from firm 2) then the optimal prices depend on the relationship between N_1 and s. If N_1 is low relative to s then only part of the market will be served while if N_1 is relatively large then a price is charged such that all consumers buy the product. Formally, the equilibria are:*

1. $N_1 < \frac{s-2\beta}{\beta} : p_1 = \frac{\beta N_1 + s}{2},\ z_1 = \frac{\beta N_1 + s}{2(s-\beta)},\ \pi_1 = \frac{(\beta N_1 + s)^2}{4(s-\beta)}$

2. $N_1 \geq \frac{s-2\beta}{\beta} : p_1 = \beta N_1 + \beta,\ z_1 = 1,\ \pi_1 = \beta N_1 + \beta$

Proof. Analyzing the three possible equilibria in stage 4 and it's price ranges shows that for $p_1 \geq \beta N_1 + s$ the revenue is zero (as no consumer buys the system) and for $p_1 \leq \beta N_1 + \beta$ the revenue is equal to p_1 as every consumers buys the system $(z_1 = 1)$. In between there is a quadratic revenue function with $\pi_1 = \frac{-p_1 + \beta N_1 + s}{s - \beta} p_1$ that has a unique maximum at $p_1 = \frac{\beta N_1 + s}{2}$. This is always smaller than $\beta N_1 + s$ such that equilibrium 3 $(z_1 = 0)$ is never played. Playing equilibrium 2 $(z_1 = 1)$ is optimal if the maximum point of the quadratic function is outside it's price region. Then, the optimal price for firm 1 is right at the border with $p_1 = \beta N_1 + \beta$ as the combined revenue function at that point is downward sloping both to the left (linear, equilibrium 2) and to the right (quadratic, equilibrium 1). This is the case if $\frac{\beta N_1 + s}{2} \leq \beta N_1 + \beta$ which leads to $N_1 \geq \frac{s-2\beta}{\beta}$. It is of course intuitive that the stronger the already existing network the more likely it is that it is profitable for firm 1 to serve the whole market, holding the other parameters constant. □

Proposition 4. *If the incumbent is a monopolist in this stage selling quality* 1 *to* *the consumers (it's own technology) then it is always optimal to serve the whole market. Formally:* $p_1 = \beta N_1 + \beta$, $z_1 = 1$, $\pi_1 = \beta N_1 + \beta$.

Proof. The proof is similar to the one above but here a price to serve only part of the market is only optimal for $N_1 < \frac{1-2\beta}{\beta} < 0$ as $\frac{1}{2} < \beta < 1$. Therefore for any $N_1 > 0$ it is always optimal to serve the full market if quality is 1. □

Proposition 5. *If both firms are in the market in stage 3 selling their own systems to the consumers then the optimal prices again depend on the relationship between* N_1 *and* s*. If* N_1 *is relatively low then there is no equilibrium in real strategies. For high* N_1 *the incumbent with the lower product quality can keep the entrant out of the market and sell it's product to all consumers. For values of* N_1 *that are in between both firms sell their systems and the full market is served. Formally:*

1. $N_1 \geq \frac{2s-3\beta-2}{\beta}$: $p_2 = z_2 = \pi_2 = 0$, $p_1 = \pi_1 = \beta N_1 - s + \beta + 1$, $z_1 = 1$

2. $\underline{N_1} \leq N_1 < \frac{2s-3\beta-2}{\beta}$: $(p_1, p_2) = (\frac{\beta N_1+s-3\beta-1}{3}, \frac{-\beta N_1+2s-3\beta-2}{3})$
   $(\pi_1, \pi_2) = (\frac{(\beta N_1+s-3\beta-1)^2}{9(s-2\beta-1)}, \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)})$

3. $N_1 < \underline{N_1}$: *No equilibrium in pure strategies*

For the minimal N_1 *for which a pure strategy price equilibrium is possible* $\frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)} \leq \underline{N_1} \leq \frac{s-3\beta-1}{2\beta}$ *has to hold.*

Proof. The formal proof is shown in the appendix. Note that the price equilibria lead to a situation such that only consumer equilibria 1b (market sharing, full market served) or 2b (only system 1 is sold to the full market) are played in stage 4. The reason is due to the fact that if consumer equilibria 2a, 2b, 3a, 3b and 4 are played at least one of the two firms does not sell anything. Therefore, firms will choose a different price whenever possible. The 2b equilibrium therefore only appears for high values of N_1 for which the second firm cannot set another positive price to do better given firm 1's optimal price (see appendix). Such a situation however does not exist for the other consumer equilibria in this group. A price vector such that consumer equilibrium 1a will be played does not exist because due to the relatively strong network effect it is never optimal for firm 1. Setting a price such that equilibrium 1b (market sharing) is played is optimal for a relatively wide range of values of N_1 but it breaks down for very low values such that no equilibrium exists there. □

5.3.2. Stage 2

Using the results of stage 3 one can derive a condition to entry for firm 2 in stage 2:

Proposition 6. *If the existing network for firm 1 and the entry cost for firm 2 are relatively small then it is optimal for firm 2 to enter the market, otherwise it is optimal to stay out. Formally,* $N_1 \leq \frac{2s-3\beta-2-3\sqrt{F_2(s-2\beta-1)}}{\beta}$ *has to hold such that firm 2 enters the market.*

Proof. Given that in stage 3 a value of $N_1 \geq \frac{2s-3\beta-2}{\beta}$ leads to zero profits for firm 2 it can never be optimal to enter in that case. If N_1 is lower, then there is an equilibrium in stage 3 that leads to market sharing in stage 4 such that the revenue is positive and equal to $\pi_2 = \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$ (see Proposition 5). This revenue has to be greater than the fixed cost and solving $F_2 \leq \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$ for N_1 leads to the expression above. □

Note that due to this entry condition the only equilibrium that will be played in the duopoly case in stage 4 is the 1b equilibrium as for the 2b equilibrium firm 2 would not enter the market. The result shows that N_1 and F_2 make firm 1 tougher in a sense that a high value deters entry. This is comparable to the general approach taken by Fudenberg and Tirole (1984) about over- or underinvestment (in a very general meaning) deterring entry by making the incumbent tough in the following stages. In this model there is no endogenous investment but N_1 and F_2

are exogenous variables that however have the same effect as the investments in the classification by Fudenberg and Tirole.

5.3.3. Stage 1

The analysis of the equilibria in stage 1 depends on the entry decision in stage 2:

Proposition 7. *If firm 2 cannot enter the market in stage 2, agreeing on a licensing agreement in stage 1 is only optimal for both firms if the existing network and the compatibility cost for firm 1 are relatively small:* $N_1 < \frac{s-2\beta}{\beta}$ *and* $F_1 \leq \frac{(\beta N_1+s)^2}{4(s-\beta)} - \beta N_1 - \beta$*. Furthermore, the no-entry condition (*$N_1 > \frac{2s-3\beta-2-3\sqrt{F_2(s-2\beta-1)}}{\beta}$*) and the first inequality do not contradict each other only for* $F_2 > \frac{(s-\beta-2)^2}{9(s-2\beta-1)}$*. For the licensing fee* $0 \leq T \leq \frac{(\beta N_1+s)^2}{4(s-\beta)} - \beta N_1 - \beta - F_1$ *holds.*

Proof. As firm 2 cannot enter the market in stage 2 licensing is always optimal for any $T \geq 0$. By licensing the better technology firm 1 can earn $\beta N_1 + \beta$ if $N_1 \geq \frac{s-2\beta}{\beta}$ and $\frac{(\beta N_1+s)^2}{4(s-\beta)}$ if $N_1 < \frac{s-2\beta}{\beta}$ (see Proposition 3) and $\beta N_1 + \beta$ by staying monopolist with it's own technology (see Proposition 4). The difference in those revenues has to be larger than the compatibility cost $F_1 > 0$ plus the licensing fee $T geq 0$. Clearly, for $N_1 \geq \frac{s-2\beta}{\beta}$ the difference in revenues is zero such that this is not possible. Licensing can therefore only be optima for a smaller N_1 and F_1 can only be as large as the difference in revenues. The licensing fee that the two firms agree on is then between zero and the revenue difference minus compatibility cost which leads to the inequality above. □

For the situation when firm 2 can enter the market the following lemma is needed:

Lemma 1. *For a wide range of values of existing network* N_1 *the revenue for a monopolist that sells products with quality* s *is greater than the combined industry* *revenues of a duopoly when one firm sells quality s and the other firm quality* 1 *and this can be shown for $N_1 \geq \frac{s-3\beta-1}{2\beta}$. Althoughthis is not the minimal N_1 for which the lemma holds it cannot be shown for other values $\underline{N_1} \leq N_1 \leq \frac{s-3\beta-1}{2\beta}$.*

Proof. The revenue for the monopolist is $\beta N_1 + \beta$ if $N_1 \geq \frac{s-2\beta}{\beta}$ and $\frac{(\beta N_1+s)^2}{4(s-\beta)}$ if $N_1 < \frac{s-2\beta}{\beta}$ while the combined industry revenue is $\frac{(\beta N_1+s-3\beta-1)^2+(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$ (see Propositions 3 to 5). Note that for $N_1 < \frac{s-2\beta}{\beta}$, $\beta N_1+\beta$ is a lower bound for the actual revenue function $\frac{(\beta N_1+s)^2}{4(s-\beta)}$. Comparing prices shows that the monopoly price $\beta N_1+\beta$ is greater than both duopoly prices as long as $N_1 > \frac{s-3\beta-1}{2\beta}$. As in both cases the full market is served also the monopoly revenue has to be always greater. This is the case for $N_1 \geq \frac{s-2\beta}{\beta}$ but also for $\frac{s-3\beta-1}{2\beta} \leq N_1 < \frac{s-2\beta}{\beta}$ as $\frac{(\beta N_1+s)^2}{4(s-\beta)} > \beta N_1 + \beta$ for $N_1 < \frac{s-2\beta}{\beta}$. One can also assume that the monopoly profit is greater for at least some values in $\underline{N_1} < N_1 < \frac{s-3\beta-1}{2\beta}$ as $\beta N_1+\beta$ is still greater than the lower duopoly price and the actual revenue $\frac{(\beta N_1+s)^2}{4(s-\beta)}$ is even higher than $\beta N_1 + \beta$. To calculate this one has to however check the revenue functions directly which leads to the inequality $\frac{(\beta N_1+s)^2}{4(s-\beta)} > \frac{(\beta N_1+s-3\beta-1)^2+(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$ which cannot be reasonably solved for a general N_1. □

Proposition 8. *If firm 2 can enter the market in stage 2, agreeing on a licensing agreement is optimal for both firms if the difference between the compatibility cost and the market entry cost is not too large and it is always optimal if the market entry cost is larger than the compatibility cost. Formally:*

1. $N_1 \geq \frac{s-2\beta}{\beta}$: $F_1 - F_2 \leq \beta N_1 + \beta - \frac{(\beta N_1+s-3\beta-1)^2+(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$

2. $N_1 < \frac{s-2\beta}{\beta}$: $F_1 - F_2 \leq \frac{(\beta N_1+s)^2}{4(s-\beta)} - \frac{(\beta N_1+s-3\beta-1)^2+(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$

Proof. If firm 2 can enter the market in stage 2 then $\pi_2 - F_2 > 0$ and the conditions $\pi^L - \pi_1 \geq T + F_1$ and $\pi_2 - F_2 \leq T$ can be combined to $\pi^L - (\pi_1 + \pi_2) \geq F_1 - F_2$: The difference between the monopoly revenue to the total duopoly revenues has to be larger than the difference of the compatibility cost for firm 1 and the entry cost for firm 2. Due to Lemma 1 for $N_1 \geq \frac{s-3\beta-1}{2\beta}$ the left hand side is positive such that licensing is always optimal if the compatibility cost for firm 1 is lower than the entry cost for firm 2. Furthermore, it is also optimal whenever the difference between the two is small enough. Plugging in the revenue functions from Proposition 3 to 5 leads to the inequalities shown above. □

5.3.4. Summary

The analysis shows that there are two ways such that licensing can be optimal: When firm 2 cannot enter the market it is still optimal when firm 1's existing network and the compatibility cost are relatively low. The intuition behind this result is that for a large existing network N_1 firm 1 is not better off by being a monopolist with the better technology. The compatibility cost has to be lower than the difference in the monopoly profits for the two technologies. On the other hand when firm 2 can enter the market licensing is optimal when the compatibility cost is not much larger than the market-entry cost or even lower. Then the firms will always agree to license the technology (this can only be shown for $N_1 \geq \frac{s-3\beta-1}{2\beta}$ though).

There are therefore three possible equilibria in the complete game: Both firms agree on a licensing contract in stage 1 and then firm 1 acts as a monopolist in stage 3 to either serve the full market if the existing network is high or serve only part of the market otherwise. When there is no licensing and firm 2 does not enter then the same is true in stages 3 and 4 although then firm 1 always serves the full market and makes less profits when the existing network is small. If firm 2 does enter however in stage 2 then there is a duopoly competition in prices in stage 3 and consumers play equilibrium 1b (full market served and market sharing) in stage 4. Unfortunately, in this model the results break down for low values of N_1 as then there is no pure strategy duopoly equilibrium in stage 3 and then revenues cannot be compared between the different strategies.

Going back to the Apple vs Microsoft story it might seem that licensing would have been a good strategy for Apple as it was able to enter the market and then the model suggests that licensing is almost always optimal. The compatibility cost however might have been too high compared to the entry cost but in reality Microsoft wanted Apple to open up it's technology so that is not a good explanation. The question of course is whether Apple did even consider licensing before entering the market. In that case this model does not explain the situation very well as in a sense Apple restricted it's strategy space and basically jumped over the decision in stage 1 to just decide whether to enter or not. In this circumstance the model (from stage 2 onwards) is only able to explain the market sharing after the entry decision when Apple was only able to gain a small market share (with high revenues however) due to the existing software network of Microsoft. What also has to be taken into account is that Apple not only sold their operating system but also hardware. To fit reality better one would have to introduce a factor into the revenue equations that reflects the fact that firm 2 also has hardware sales while firm 1 does not. In that case licensing is less likely due to the fact that the second firm also loses some of it's hardware revenues when it agrees to a licensing contract. Furthermore, the game at hand is a one-shot game which does not take into account dynamic effects such that firm 2 could build up a strong network over time which might have been a strategy for Apple as well. If that was the case then it was clearly not a successful one as Microsoft was able to catch up in quality and Apple was never able to challenge Microsoft's network.

5.4. Open Source: Market Entry

Another topic that can be discussed using the model is open source software. Although the topic is very broad I want to show that some parts can be modeled in this framework. Using the studies by Lerner and Tirole, I take three classes of arguments and try to capture them with the model at hand. The first one is the argument that software might be made open source whenever it is probable that the software would not be profitable on the market. The second class of arguments are the signaling incentives like future job opportunities and access to the capital market. A third argument is an external revenue stream that makes it profitable to give away the main product for free. Furthermore, the authors expect that open source software brings down revenues of closed source software and I will also discuss whether that is true in this framework.

To use the network type model for general software one of course has to discuss how a network is built in the model. For the operating systems it is clear that the network is software that is written and sold for the respective system. This gives additional utility for consumer and this is greater the greater the network size is. This is also true for other types of software like middleware as it is the basis for other software. A more abstract and general notion of network is to assume that the network property holds true also for other types of software, due to learning effects, "private support" by other users or complementary products like other software or documentation. Communication software is also more useful the more people use the software (like the general telephone network). I assume here that this is the case for any software I discuss. Certainly, different types of software have different degrees of network effects and for some this might be negligible. A detailed discussion of those features of software would of course need a separate study and is not part of this work.

The idea to make software open source software whenever the product would not be profitable of course is just one of many ideas to justify open source but it is one that is consistent with the revenue maximizing assumption in economic theory. Using the same model as the one for operating systems licensing, I assume that the products now are software packages that have a distinct quality. Note further that there is no second firm that makes decisions on entry and prices but only a software package with quality s that is available to the consumers at $p_2 = 0$.

Proposition 9. *The condition for open source is* $N_1 > \frac{2s-3\beta-2-3\sqrt{F_2(s-2\beta-1)}}{\beta}$. *In that case there are three different equilibria in stage 3 that lead to different consumer equilibria in stage 4: For low values of* N_1 *the open source product takes over the full market and for high values the open source product is not sold at all. In between the market is shared between the incumbent and the open source software* *and the full market is served in all three equilibria:*

1. $N_1 \leq 1$: $(p_1, p_2) = (0, 0)$, $(z_1, z_2) = (0, 1)$

2. $1 < N_1 < \frac{2s-3\beta-2}{\beta}$: $(p_1, p_2) = (\frac{\beta N_1 - \beta}{2}, 0)$, $(z_1, z_2) = (\frac{\beta N_1 - \beta}{2(s-2\beta-1)}, \frac{-\beta N_1 + 2s - 3\beta - 2}{2(s-2\beta-1)})$

3. $N_1 \geq \frac{2s-3\beta-2}{\beta}$: $(p_1, p_2) = (\beta N_1 - s + \beta + 1, 0)$, $(z_1, z_2) = (1, 0)$

The equilibrium when firm 2 does not release the software as open source but enters the market in stage 2 is equivalent to the one for the operating systems licensing case. Comparing the price vectors for the open source strategy and the market entry strategy shows that open source prices are always lower (even for firm 1) if $N_1 < \frac{2s-3\beta-2}{\beta}$ *and equal if* $N_1 \geq \frac{2s-3\beta-2}{\beta}$.

Proof. The condition for open source is the same as the market entry condition from the licensing case above. If firm 2 cannot profitably enter the market in stage 2 then releasing the software as open source also maximizes profits to zero[33]. In stage 3 firm 1 can choose between prices that lead to equilibria 1b, 2b and 3b in the consumer stage given that $p_2 =$ holds. The optimal price for the quadratic revenue function for the 1b equilibrium is $p_1 = \frac{\beta N_1 - \beta}{2}$. As $p_1 \geq 0$ has to hold the optimal price however is $p_1 = 0$ if $N_1 \leq 1$ (3b). Furthermore, if $N_1 \geq \frac{2s-3\beta-2}{\beta}$ then the maximum is in the range of the 2b equilibrium and it is therefore optimal to choose the border value at $p_1 = \beta N_1 - s + \beta + 1$ (2b). This 2b equilibrium is the same as the one from the licensing case. If $1 < N_1 < \frac{2s-3\beta-2}{\beta}$ then it is optimal to choose $p_1 = \frac{\beta N_1 - \beta}{2}$ (1b). Comparing prices clearly shows that both p_1 and p_2 are always smaller than when firm enters the market. □

[33] I assume that licensing is not an option. If it was then either $N_1 \geq \frac{s-2\beta}{\beta}$ or $F_1 \leq \frac{(\beta N_1 + s)^2}{4(s-\beta)} - \beta N_1 - \beta$ has to hold also such that the firm is not better off by using conventional licensing.

5.5. Open Source: Signaling Incentives

Using signaling incentives in the model requires a change in perception of the second firm which now does not have the goal to necessarily maximize revenues but has other incentives like future job opportunities and access to capital market. In this case it is probably useful to not think of firm 2 as a standard firm but rather a private initiative, a research group at a university, a small startup or a hobby programmer. Lerner and Tirole show that this is a very common way how open source products start it's business (Lerner and Tirole, 2002). The crucial assumption I make is that those signaling incentives are maximized when the network of the product (or the number of buyers) are maximized for firm 2. The hypothesis behind this is that job signaling and capital market signaling are optimal whenever the product awareness is maximized and this is done by maximizing the network.

If in stage 1 the decision is made for open source licensing then $p_2 = 0$ which leads to the same stage 3 and 4 equilibria as in the market entry case. The benchmark however is not the profit maximizing case but the case where the entrant maximizes z_2 in stage 3 and not the profit, given that at least the entry cost F_2 can be earned:

Proposition 10. *If firm 2 maximizes the network size instead of revenues in stage 3 then for a low N_1 firm 2 can set a positive price and earn positive revenues by taking over the full market while the other two equilibria are the same as above (market entry). Therefore, both $N_1 \leq 1$ and $F_2 \leq -\beta N_1 + \beta$ have to hold such that entry is possible. Prices in the open source case are again smaller as firm 2 has to set a positive price when entering the market. To formalize the pricing stage equilibria:*

1. *$N_1 \leq 1$: $(p_1, p_2) = (0, -\beta N_1 + \beta)$, $(z_1, z_2) = (0, 1)$*

2. *$1 < N_1 < \frac{2s-3\beta-2}{\beta}$: $(p_1, p_2) = (\frac{\beta N_1 - \beta}{2}, 0)$, $(z_1, z_2) = (\frac{\beta N_1 - \beta}{2(s-2\beta-1)}, \frac{-\beta N_1 + 2s - 3\beta - 2}{2(s-2\beta-1)})$*

3. $N_1 \geq \frac{2s-3\beta-2}{\beta}$: $(p_1, p_2) = (\beta N_1 - s + \beta + 1, 0)$, $(z_1, z_2) = (1, 0)$

Proof. Maximizing z_2 in stage 3 leads to either $p_2 = 0$ or $p_2 > 0$ whenever $z_2 = 1$ already holds. The latter happens when $p_2 = -\beta N_1 + \beta$ and then for any $p_1 > 0$ a 3b consumer equilibrium is played which already maximizes z_2. Of course $N_1 \leq 1$ has to hold as $p_2 \geq 0$. This is the first equilibrium for $N_1 \leq 1$. If $N_1 > 1$ network maximization is done by setting $p_2 = 0$ and the equilibria are the same as the ones for the market entry case. Clearly, in case of signaling incentives entering is not optimal when $N_1 > 1$ because for $F_2 > 0$ the profit is negative while it is zero in the open source case. If $N_1 \leq 1$ then for $F_2 \leq -\beta N_1 + \beta$ the profit for market entry is non-negative such that open source is not optimal in that case. □

Remark: The conditions for entry mean that both N_1 and F_2 have to be quite small which is equivalent to saying that entry has to be easy. In that case a private initiative or a startup might enter the market with it's superior product but whenever entry is harder then open source is the way to go.

5.6. Open Source: External Revenue

The third aspect about open source is that firms can earn off open source software by offering service contracts, additional software and services. This is the route many open source companies go and it also fits the strategy Google is using: Google offers software and services for free (not always open source though) and makes money by selling advertising. The idea behind this is that selling advertising leads to a higher profit than selling the actual product. Of course, a company could both sell the commercial software and use advertising as well but this is not common and might be rejected by consumers.

To model this strategy I assume that in stage 1 firm 2 can opt to enter the market in a conventional way and compete or release it as open source. This enforces

$p_2 = 0$ but firm 2 still gets revenues of $\pi_2 = \alpha z_2$. The assumption is that the profit is linearly increasing with the network size and α is the profit per network size.

Proposition 11. *If firm 2 makes it's development open source then firm 2 takes over the full market if firm 1's existing network is low, it has to share the market for intermediate values and cannot enter for high values N_1. Formally, the stage 3 equilibria look like this:*

1. $N_1 \leq 1$*:* $(p_1, p_2) = (0, 0)$*,* $(z_1, z_2) = (0, 1)$

2. $1 < N_1 < \frac{2s-3\beta-2}{\beta}$*:* $(p_1, p_2) = (\frac{\beta N_1-\beta}{2}, 0)$*,* $(z_1, z_2) = (\frac{\beta N_1-\beta}{2(s-2\beta-1)}, \frac{-\beta N_1+2s-3\beta-2}{2(s-2\beta-1)})$

3. $N_1 \geq \frac{2s-3\beta-2}{\beta}$*:* $(p_1, p_2) = (\beta N_1 - s + \beta + 1, 0)$*,* $(z_1, z_2) = (1, 0)$

Proof. The proof is equivalent to the first part if Proposition 9 as again $p_2 = 0$ and firm 1 optimizes profits given that price. The difference between those two cases is that firm 2's revenues are not zero. □

Proposition 12. *Open Source is an optimal strategy for firm 2 if the external revenue parameter α is large enough to compensate for the zero price. The conditions are $\alpha \geq \frac{2(-\beta N_1+2s-3\beta-2)}{9} - F_2\frac{2(s-\beta-1)}{-\beta N_1+2s-3\beta-2}$ for max$\{\underline{N_1}, 1\} < N_1 < \frac{2s-3\beta-2}{\beta}$ and $\alpha \geq \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)} - F_2$ for $\underline{N_1} \leq N_1 \leq 1$ and $\underline{N_1} < 1$. This includes the case where firm 2 could not enter the market anyways.*

Proof. The last argument can be shown easily by using the same proof as in Proposition 9: When the alternative (enter the market) does not offer positive profits then open source is at least as good a strategy for firm 2. Comparing the market entry condition from Proposition 6 ($F_2 \leq \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}$) to the conditions from above shows that those are always fulfilled for any $\alpha \geq 0$ when the market entry condition is not met. The conditions stated above can be derived by comparing the profit functions for the two cases. For open source those profits are $\Pi_2 = \alpha$ for $N_1 \leq 1$, $\Pi_2 = \alpha\frac{-\beta N_1+2s-3\beta-2}{2(s-2\beta-1)}$ for $1 < N_1 < \frac{2s-3\beta-2}{\beta}$ and $\Pi_2 = 0$ for $N_1 \geq \frac{2s-3\beta-2}{\beta}$. For market entry the profits are $\Pi_2 = -F_2$ for $N_1 \geq \frac{2s-3\beta-2}{\beta}$ and

$\Pi_2 = \frac{(-\beta N_1+2s-3\beta-2)^2}{9(s-2\beta-1)}) - F_2$ for $\underline{N_1} \leq N_1 < \frac{2s-3\beta-2}{\beta}$. The comparisons lead to the conditions shown above for the different ranges of N_1. □

Note that there is again no result for $N_1 < \underline{N_1}$ because then there is no pure strategy equilibrium in case of entry (duopoly situation in stage 3) and therefore no benchmark value. The results show that open source can be optimal from a profit standpoint even when firm 2 could enter the market and earn a positive profit. This is due to two effects: By releasing the software as open source the software is more interesting for consumers than it would be otherwise which leads to a higher market share and higher profits. The second effect is that the firm also does not have to pay the entry cost. Depending on the other parameters α can therefore be a lot smaller than the competitive price for firm 2 and still make open source optimal.

5.7. Other Topics and Remarks

The last sections showed some of the topics that can be discussed using the developed model. However, there are many more related topics that could have also been discussed but the scope of this work has to be limited to a certain extent. Here, I want to just shortly show and discuss some of those topics:

Related to the licensing topic is the topic about smartphone systems and the system competition that started a few years ago and is still ongoing (see above). One could assume Apple as the incumbent and Google's Android system as the entrant. However, there are a few differences to the operating systems case, most notably that Apple is probably still the technological leader and that Google is also using an open source strategy. This would need a model with qualities $s_2 < s_1$ and $p_2 = 0$.

Another topic that can be analyzed is the current "codec war" between the closed MPEG codecs and Google's open WebM format. The network in this case is the number of videos coded in a certain format which gives utility to the consumers. Codecs are of course not directly sold to consumers but are installed in media players or web browsers that consumers install. Again, in this situation $s_2 < s_1$ and $p_2 = 0$ has to hold.

One of the topics of openness that was completely neglected was the issue of compatibility. This touches the topics of middleware products which enforce software compatibility of the underlying operating systems because software can always run on any system. Clearly, this could be analyzed by using the operating systems licensing approach and compare the equilibria with and without middleware. However, this is part of a general topic that is already discussed in the literature a lot (see related work) and therefore the other topics seem more interesting for this work.

The overall analysis however lacks a few aspects that I want to mention here and that could be a starting point for future work:

- I already stated that the analysis for open source is hardly complete by just implementing some different incentives and revenues into a model. Note also that in this type of model there is no distinction between open source or free software. For all those incentives providing the software for free is as good as providing it as open source. Open source however seems to have more benefits both for the developer and for the consumers.

- The analysis of licensing is somewhat incomplete whenever the competing companies act in vertically integrated markets as is the case in the Apple/Microsoft story.

- The model is not solved in a general way but for certain parameter combinations and the model further allows for more complex setups, for example using different network parameters for different firms (maybe a way to go in the open source discussion) and using different ranges of parameter values. The model was solved for combinations that keep the possible pricing stage equilibria at a minimum and also ensure unique consumer stage equilibria. That there is no pure strategy equilibrium for small values of N_1 in the duopoly case (stage 3) could possibly be overcome by discussing different parameter ranges.

- In a more general perspective the model can be seen as a short term model for the situation in which the game is over after one licensing and one pricing decision. The setup of the model however could be used to play the game repeatedly which would allow for more complex strategies and also for an endogenous explanation of the lead of the incumbent (parameter N_1).

6. Summary, Conclusions and Possible Extensions

The thesis shows that one can indeed formulate openness as a general term for a class of problems that can often be found especially in the sector of information technology. This includes open source, compatibility issues and licensing. The approach to discuss the topics in general offers an interesting view and seems useful. That openness is treated very differently by different firms but also within the firms is the result of the first part of this thesis: While Google is open in almost any possible way (except it's search technology), Microsoft traditionally is not. Although becoming more open recently this has to be attributed at least to a certain extent to pressure from antitrust courts both in the US and in Europe. Apple did not face the same pressure and has to be considered even less open in some ways (operating system, hardware platform, iTunes, iPhone and others) although more open in others which is shown in some open source projects to which Apple is contributing to.

The economic analysis in the second part of the thesis shows how certain topics can be discussed using a model that features network economics, vertical product differentiation and an incumbent-entrant scheme. Using it for a licensing topic shows how licensing is often optimal whenever the incumbent faces competition due to the entry of another company with a better technology. The same does not hold true as strongly whenever the competitor could not enter anyways (for example due to high entry costs) although there are some parameter combinations that still would make licensing optimal. Using the model for open source issues one can show how certain incentives that are mentioned in the literature lead to situations where open source is optimal for the potential entrant: Whenever entry would not be profitable, whenever there are signaling incentives involved and whenever there is an external source of revenue for the firm.

The thesis however only covers a small subset of issues and topics that can be discussed, to mention only a few more: Compatibility was not discussed in that setup and there is also the topic of standards that was neglected. For several other topics a more sophisticated model would be useful and a few suggestions were made in the last section. I also only discussed the abstractions and assumptions of the model in a rather basic form and a detailed study about the industry possibly using empirics and econometrics would be beneficial to the overall analysis.

7. References

Bibliography

Arthur, W Brian (1989), “Competing Technologies, Increasing Returns, and Lock-In by Historical Events.” *Economic Journal*, 99, 116–131.

Baake, Pio and Anette Boom (2001), “Vertical product differentiation, network externalities, and compatibility decisions.” *International Journal of Industrial Organization*, 19, 267–284.

Campbell-Kelly, Martin (2001), “Not Only Microsoft: The Maturing of the Personal Computer Software Industry, 1982-1995.” In *Computers and Communications Networks*, volume 75 of *The Business History Review*, 103–145, The President and Fellows of Harvard College.

Church, Jeffrey and Neil Gandal (1992), “Network Effects, Software Provision, and Standardization.” *Journal of Industrial Economics*, 40, 85–103.

Cortada, James W. (2002), “Researching the History of Software from the 1960s.” *IEEE Annals of the History of Computing*, 24, 72–79.

Davis, Steven J. and Kevin M. Murphy (2000), “A Competitive Perspective on Internet Explorer.” *American Economic Review*, 90, 184–187.

Fisher, Franklin M. (2000), “The IBM and Microsoft Cases: What’s the Difference?” *American Economic Review*, 90, 180–183.

Fudenberg, Drew and Jean Tirole (1984), “The Fat-Cat Effect, the Puppy-Dog Ploy, and the Lean and Hungry Look.” *American Economic Review*, 74, 361–66.

Gilbert, Richard J. (1999), “Networks, Standards, and the Use of Market Dominance: Microsoft (1995).” In *The Antitrust Revolution: Economics, Competition and Policy (Oxford 1999),*, 409–429, Oxford University Press.

Goodell, Jeff (1996), “The Rise and Fall of Apple Computer.” *Rolling Stone*, 731,732.

Haigh, Thomas (2001), “Inventing Information Systems: The Systems Men and the Computers, 1950-1968.” In *Computers and Communications Networks*, volume 75 of *The Business History Review*, 15–61, The President and Fellows of Harvard College.

John, Richard R. (2001), “Rendezvous with Information? Computers and Communications Networks in the United States: [Introduction].” In *Computers and Communications Networks*, volume 75 of *The Business History Review*, 1–13, The President and Fellows of Harvard College.

Lerner, Josh and Jean Tirole (2002), “Some Simple Economics of Open Source.” *Journal of Industrial Economics*, 50, 197–234.

Lerner, Josh and Jean Tirole (2005), “The Economics of Technology Sharing: Open Source and Beyond.” *Journal of Economic Perspectives*, 19, 99–120.

Linzmayer, Owen W. (2004), *Apple Confidential 2.0 The Definitive History of the World’s Most Colorful Company*, 2 edition. No Starch Press.

Moore, Gordon E. (1965), “Cramming more components onto integrated circuits.” *Electronics Magazine*, 38.

Shy, Oz (2004), *The Economics of Network Industries*, 2 edition. Cambridge University Press.

Vise, David A. (2008), *The Google Story*, 2 edition. Pan.

Välimäki, Mikko (2010), “Two types of openness in information technology standards and competition policy.” *Open standards and antitrust*, 13–38.

Weblinks

WordNet 3.0. Princeton University, n.d. http://wordnet.princeton.edu. 2011/07/27.

Intel. *Moore's Law*, n.d. http://www.intel.com/technology/mooreslaw. 2011/07/27.

Globalfoundries. *GLOBALFOUNDRIES Breaks Ground on World's Most Advanced Semiconductor Foundry*, 2009. http://www.globalfoundries.com/newsroom/2009/20090724.aspx. 2011/07/27.

Jason Pontin. *On Openness.* Technology Review, 2009. http://www.technologyreview.com/web/23170. 2011/07/27.

Open Source Initiative. *Open Source Definition*, n.d. http://www.opensource.org/docs/osd. 2011/07/27.

Jim Mooney. *Software Portability Home Page*, 2008. http://www.cs.wvu.edu/~jdm/research/portability/home.html. 2011/07/27.

Microsoft. *Key Events in Microsoft History*, 2004. http://download.microsoft.com/download/7/e/a/7ea5ca8c-4c72-49e9-a694-87ae755e1f58/keyevents.doc. 2010/07/26.

The History of Computing Project. *Microsoft Company 15 September 1975*, n.d. http://www.thocp.net/companies/microsoft/microsoft_company.htm. 2010/07/28.

Andy F. Mesa. *The Apple Museum*, n.d. http://applemuseum.bott.org. 2010/09/04.

Glen Sanford. *apple-history.com*, n.d. http://www.apple-history.com/. 2010/09/22.

Google. *Google Corporate Timeline*, n.d.a. http://www.google.com/intl/en/corporate/history.html. 2010/09/26.

Operating System Documentation Project. *Windows Family*, n.d. `http://www.operating-system.org/betriebssystem/_english/bs-windows.htm`. 2010/07/28.

OpenOffice.org. *OpenOffice.org History and Milestones 1999 - 2005*, n.d. `http://www.openoffice.org/about_us/milestones.html`. 2010/07/29.

HPC:Factor. *The History of Microsoft Windows CE*, n.d. `http://www.hpcfactor.com/support/windowsce`. 2010/07/29.

Peter Bright. *Microsoft embedded roadmap has iPad-shaped hole in it.* Ars Technica, n.d. `http://arstechnica.com/microsoft/news/2010/06/microsofts-embedded-roadmap-clear-in-places-murky-in-others.ars`. 2010/07/29.

Robert Jaques. *Windows CE knocks Palm OS off the top spot.* V3.co.uk, 2004. `http://answers.softpicks.net/answers/topic/Windows-CE-surpassed-Palm-OS-in-marketshare--9646-1.htm`. 2010/07/29.

The GigaOM Network. *iPhone vs Others: The Mobile OS Market*, 2010. `http://gigaom.com/2010/03/18/the-mobile-os-market`. 2010/07/29.

Microsoft. *Nokia and Microsoft Announce Plans for a Broad Strategic Partnership to Build a New Global Mobile Ecosystem*, 2011. `http://www.microsoft.com/presspass/press/2011/feb11/02-11partnership.mspx`. 2011/05/17.

Eric Sink. *Memoirs From the Browser Wars*, 2003. `http://www.ericsink.com/Browser_Wars.html`. 2010/07/30.

Scott Schnoll. *The History of Internet Explorer*, n.d. `http://www.nwnetworks.com/iehistory.htm`. 2010/07/30.

Wikipedia. *Usage share of web browsers*, n.d. `http://en.wikipedia.org/wiki/Usage_share_of_web_browsers`. 2010/07/30.

IEBlog. *The Windows Internet Explorer Weblog*, n.d. `http://blogs.msdn.com/b/ie`. 2010/07/30.

Shannon Blogs. *MSN Homepages throughout Time*, 2000. http://msnblog.msn.com/blogpost.aspx?post=1350394. 2010/08/03.

Dick Craddock. *A short history of Hotmail.* Windows Live Blogs, n.d. http://windowsteamblog.com/windows_live/b/windowslive/archive/2010/01/06/a-short-history-of-hotmail.aspx. 2010/08/03.

Jeff Kunins. *Windows Live Messenger – a short history.* Windows Live Blogs, 2010. http://windowsteamblog.com/windows_live/b/windowslive/archive/2010/02/09/windows-live-messenger-a-short-history.aspx. 2010/08/03.

BBC News. *Microsoft and Yahoo seal web deal*, 2009a. http://news.bbc.co.uk/2/hi/business/8174763.stm. 2010/08/03.

Experian Hitwise. *Top 20 Sites & Engines*, n.d. http://www.hitwise.com/us/datacenter/main/dashboard-10133.html. 2010/08/04.

Netmarketshare. *Search Engine Market Share*, n.d. http://marketshare.hitslink.com/search-engine-market-share.aspx?qprid=4&qptimeframe=M. 2010/08/04.

Marc Liron. *A Little Windows Media Player History...*, n.d. http://www.updatexp.com/windowsxpmediaplayer.html. 2010/07/31.

Steven Lacher. *The History of Microsoft Visual Basic*, n.d. http://www.ehow.com/about_5566758_history-microsoft-visual-basic.html. 2010/07/31.

Microsoft .NET Support Team. *Microsoft Visual Studio Release History/Timelines/Milestones*, 2009. http://msdotnetsupport.blogspot.com/2009/12/microsoft-visual-studio-release.html. 2010/07/31.

Mono. n.d. http://www.mono-project.com/Main_Page. 2010/07/31.

Anjaiah Keesari. *An Introduction to Silverlight*, 2010. http://www.codeproject.com/KB/silverlight/ASilverlightIntroduction.aspx. 2010/07/31.

Coding Unit. *The History of DirectX*, n.d.a. http://www.codingunit.com/the-history-of-directx. 2010/07/31.

Coding Unit. *The History of OpenGL*, n.d.b. http://www.codingunit.com/the-history-of-opengl. 2010/07/31.

thegameconsole.com. *A Brief History of the Home Video Game Console*, n.d. http://www.thegameconsole.com. 2010/07/31.

VGChartz. *Hardware Totals*, n.d. http://www.vgchartz.com/hardware_totals.php. 2010/07/31.

Peter Robins. *Microsoft's Zune: history of misfortune.* Guardian News and Media Limited, 2009. http://www.guardian.co.uk/technology/2009/may/26/microsoft-zune-history. 2010/08/03.

Mary Jo Foley. *Adios, Kin. We hardly knew you.* ZDNet, 2010. http://www.zdnet.com/blog/microsoft/adios-kin-we-hardly-knew-you/6719. 2010/08/03.

Wired.com. *U.S. v. Microsoft: Timeline*, 2002. http://www.wired.com/techbiz/it/news/2002/11/35212. 2010/08/06.

Department of Justice - Antitrust Devision. *United States v. Microsoft*, n.d. http://www.justice.gov/atr/cases/ms_index.htm. 2010/08/11.

Lamia Abu-Haidar. *Microsoft investigated in Europe.* CNET News, 1993. http://news.cnet.com/2100-1023_3-204317.html. 2010/08/16.

Declan McCullagh. *EU looks to wrap up Microsoft probe.* CNET News, 2002. http://news.cnet.com/2100-1001_3-941090.html. 2010/08/16.

Michael Parsons and Jo Best. *EU slaps record fine on Microsoft.* CNET News, 2004. http://news.cnet.com/EU-slaps-record-fine-on-Microsoft/2100-1001_3-5178281.html. 2010/08/16.

Commission of the European Communities. *Commission Decision of 24.03.2004 relating to a proceeding under Article 82 of the EC Treaty (Case COMP/C-3/37.792 Microsoft)*, 2002. http://ec.europa.eu/comm/competition/antitrust/cases/decisions/37792/en.pdf. 2010/08/16.

David Lawsky and Sabina Zawadzki. *EU fines Microsoft $357.3 million for defiance*, 2006. http://www.zeropaid.com/bbs/showthread.php/38334-EU-fines-Microsoft-357.3-million-for-defiance?s=95f1056083cec304d8d2f28544f4e21d. 2010/08/17.

Court of Justice of the European Union. *Judgment of the Court of First Instance (Grand Chamber) of 17 September 2007 - Microsoft v Commission (Case T-201/04)*, 2007. http://curia.europa.eu/jurisp/cgi-bin/form.pl?lang=EN&Submit=Rechercher$docrequire=alldocs&numaff=T-201/04&datefs=&datefe=&nomusuel=&domaine=&mots=&resmax=100. 2010/08/17.

BBC News. *Microsoft is accused by EU again*, 2009b. http://news.bbc.co.uk/2/hi/business/7834792.stm. 2010/08/17.

BBC News. *Microsoft offers browser choices to Europeans*, 2010a. http://news.bbc.co.uk/2/hi/technology/8537763.stm. 2010/08/17.

Nate Mook. *Microsoft Loses Appeal in South Korea*, 2006. http://www.betanews.com/article/Microsoft-Loses-Appeal-in-South-Korea/1148399316. 2010/08/17.

Bob Trott. *Microsoft pondering legal challenge to Linux*, 1998. http://www.cnn.com/TECH/computing/9811/06/linux.threat.idg. 2010/08/26.

Eric S. Raymond. *The Halloween Documents*, n.d. http://www.catb.org/~esr/halloween. 2010/08/26.

Roger Parloff. *Microsoft takes on the free world*, 2007. http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867. 2010/08/26.

Paul McDougall. *Microsoft Won't Sue Linux Users, Company Exec Says*, 2007. http://www.informationweek.com/news/global-cio/showArticle.jhtml?articleID=199501831. 2010/08/26.

Charles Babcock. *Linus Torvalds Responds To Microsoft Patent Claims*, 2007. http://www.informationweek.com/news/global-cio/showArticle.jhtml?articleID=199600443. 2010/08/26.

Dana Gardner. *Microsoft and Novell: Fox marries chicken, both move into henhouse*, 2006. http://www.zdnet.com/blog/gardner/microsoft-and-novell-fox-marries-chicken-both-move-into-henhouse/2369. 2010/08/27.

Steven J. Vaughan-Nichols. *Red Hat & Microsoft partner up*, 2009. http://blogs.computerworld.com/red_hat_microsoft_partner_up. 2010/08/27.

Andrew C. Oliver. *To Microsoft, Open Source means "Windows Encumbered"*, 2010. http://www.opensource.org/Microsoft-Open-Source. 2010/08/27.

Jim Carlton. *They Coulda Been a Contender*, 1997. http://www.wired.com/wired/archive/5.11/es_apple.html. 2010/09/10.

MG Siegler. *The Complicated Tale of AT&T's Exclusive iPhone Agreement*, 2010. http://seekingalpha.com/article/204421-the-complicated-tale-of-at-ts-exclusive-iphone-agreement. 2010/09/23.

Amy Thomson. *Verizon Wireless Said to Start Offering IPhone in January*, 2010. http://www.bloomberg.com/news/2010-06-29/verizon-wireless-said-to-start-offering-iphone-ending-at-t-s-exclusivity.html. 2010/09/23.

Nokia Blog. *Nokia and Intel create MeeGo for new era of mobile computing*, 2010. http://conversations.nokia.com/2010/02/15/nokia-and-intel-create-meego-for-new-era-of-mobile-computing. 2010/09/23.

Claudine Beaumont. *Apple iPhone 4 signal problems: Some perspective*, 2010. http://www.telegraph.co.uk/technology/apple/7894455/Apple-iPhone-4-signal-problems-Some-perspective.html. 2010/09/23.

Bill Rigby and Noel Randewich. *Jobs unveils revamped Apple TV, latest iPods*, 2010. `http://www.financialpost.com/news/Jobs+unveils+revamped+Apple+latest+iPods/3471856/story.html`. 2010/09/24.

Apple. *Apple to Use Intel Microprocessors Beginning in 2006*, 2005. `http://www.apple.com/pr/library/2005/jun/06intel.html`. 2010/09/25.

BBC News. *Apple makes Macs run Windows XP*, 2006. `http://news.bbc.co.uk/2/hi/technology/4880022.stm`. 2010/09/25.

Pingdom. *The money made by Microsoft, Apple and Google, 1985 until today*, 2010. `http://royal.pingdom.com/2010/04/09/the-money-made-by-microsoft-apple-and-google-1985-until-today`. 2010/09/25.

Apple. *Apple Reports Third Quarter Results*, 2010. `http://www.apple.com/pr/library/2010/07/20results.html`. 2010/09/25.

BBC News. *Apple passes Microsoft to be biggest tech company*, 2010b. `http://www.bbc.co.uk/news/10168684`. 2010/09/25.

Stephen Northcott. *Apple Updates App Store Developer Agreement*, 2010. `http://touchreviews.net/apple-updates-app-store-developer-agreement`. 2010/09/28.

Daniel Ionescu. *Adobe Flash Available for iPhone? Not Really*, 2009. `http://www.pcworld.com/article/173167/adobe_flash_available_for_iphone_not_really.html`. 2010/09/28.

Jeff Gamet. *EU Drops Apple iOS Developer Antitrust Investigation*, 2010. `http://www.macobserver.com/tmo/article/eu_drops_apple_ios_developer_antitrust_investigation`. 2010/09/28.

John Paczkowski. *Palm Ignores USB Group's Warning, Restores iTunes Sync*, 2009. `http://digitaldaily.allthingsd.com/20091003/palms-webos-1-2-1-restores-itunes-sync`. 2010/09/28.

Jon Buys. *Apple's Relationship to Open Source*, 2010. http://ostatic.com/blog/apples-relationship-to-open-source. 2010/10/13.

Steve Jobs. *Thoughts on Flash*, 2010. http://www.apple.com/hotnews/thoughts-on-flash. 2010/10/14.

Alastair Otter. *Only Apple can do open standards, apparently*, 2010. http://www.tectonic.co.za/2010/06/only-apple-can-do-open-standards-apparently. 2010/10/14.

K. Reed. *The Google Book Search copyright settlement and the future of information — Part 1*, 2009a. http://www3.wsws.org/articles/2009/aug2009/goog-a12.shtml. 2011/02/10.

K. Reed. *The Google Book Search copyright settlement and the future of information — Part 2*, 2009b. http://www3.wsws.org/articles/2009/aug2009/goo2-a13.shtml. 2011/02/10.

Peter Kafka. *Is YouTube Finally Ready to Turn a Profit This Year?*, 2010. http://mediamemo.allthingsd.com/20100114/is-youtube-finally-ready-to-turn-a-profit-this-year. 2011/02/10.

Chris Lefkow. *US judge tosses out Viacom copyright suit against YouTube*, 2010. http://www.google.com/hostednews/afp/article/ALeqM5h_AfErLSMMGD417l8aR0CYib0aNQ. 2011/02/10.

Google. *Transaction Fees*, n.d.b. http://market.android.com/support/bin/answer.py?answer=112622. 2011/02/15.

Eric Chu. *Android Market: a user-driven content distribution system*, 2008. http://android-developers.blogspot.com/2008/08/android-market-user-driven-content.html. 2011/02/15.

Android Market. *Accumulated number of Application and Games in the Android Market*, n.d. http://www.androlib.com/appstats.aspx. 2011/02/15.

Kevin C. Tofel. *Android Sales Overtake iPhone in the U.S.*, 2010. http://gigaom.com/2010/08/02/android-sales-overtake-iphone-in-the-u-s. 2011/05/19.

Christopher Blizzard. *Firefox, YouTube and WebM*, 2010. http://hacks.mozilla.org/2010/05/firefox-youtube-and-webm. 2011/02/14.

Chris Mills. *Opera supports the WebM video format*, 2010. http://dev.opera.com/articles/view/opera-supports-webm-video. 2011/02/14.

Dean Hachamovitch. *Another Follow-up on HTML5 Video in IE9*, 2010. http://blogs.msdn.com/b/ie/archive/2010/05/19/another-follow-up-on-html5-video-in-ie9.aspx. 2011/02/14.

Jim Bankoski. *WebM and VP8 land in Chromium*, 2010. http://blog.chromium.org/2010/05/webm-and-vp8-land-in-chromium.html. 2011/02/14.

Peter Gasston. *Playing WebM in Safari with plugins*, 2010. http://www.broken-links.com/2010/09/01/playing-webm-in-safari-with-plugins. 2011/02/14.

YouTube. *YouTube HTML5 Video Player*, n.d. http://www.youtube.com/html5. 2011/02/14.

Carlo Daffara. *An analysis of WebM and its patent risk – updated*, n.d. http://carlodaffara.conecta.it/an-analysis-of-webm-and-its-patent-risk. 2011/02/14.

Jonathan Rosenberg. *The meaning of open*, 2009. http://googleblog.blogspot.com/2009/12/meaning-of-open.html. 2011/02/14.

A. Appendix

A.1. Proof: Uniqueness of Equilibria in Consumer Stage

Equilibrium 4: Equilibrium 4 is unique as it has a unique price range where no other equilibrium is possible. Therefore, if a price vector (p_1, p_2) is a type 4 equilibrium, no other equilibrium is possible which means that a type 4 equilibrium is always unique.

Equilibrium 1a: By inspection of Table 1 one can immediately notice that equilibria 1b, 2a and 3a are impossible for the same price vector (p_1, p_2) constituting a type 1a equilibrium due to the mutually exclusive price restrictions and that a 3b type is not possible due to the different price ranges.

This leaves type 2b as the only possibility to make a type 1a equilibrium non-unique because it has both an overlapping price and a different price restriction. For 2b $p_2 \geq p_1 - \beta N_1 + s - 1 - \beta$ has to hold and for 1a $p_2 < \frac{1}{1-\beta}p_1 - \frac{\beta}{1-\beta}N_1 + s - \frac{1}{1-\beta}$. If a single price vector (p_1, p_2) constitutes both a 1a and 2b type equilibrium $p_1 - \beta N_1 + s - 1 - \beta < \frac{1}{1-\beta}p_1 - \frac{\beta}{1-\beta}N_1 + s - \frac{1}{1-\beta}$ has to hold for a feasible p_1. However, solving the inequality for p_1 leads to $p_1 > \beta N_1 + \beta$ which is not a feasible p_1 for type 2b which requires $p_1 \leq \beta N_1 + \beta$. Therefore, there is no valid combination of (p_1, p_2) that leads to both a 1a and 2b equilibrium which means that also a type 1a equilibrium is always unique.

Equilibrium 1b: Types 1a and 4 can already be excluded due to the analysis from above. Types 2b and 3b mutually exclusive price restrictions with type 1b and 2a has a different price range. For 3a and 1b to happen for the same price vector $p_1 - \beta N_1 + \beta < (s - \beta)p_1 - \beta(s - \beta)N_1 + \beta$ has to hold for a feasible p_1 such that the price restrictions for both types are met. Solving the inequality leads to $p_1 > \beta N_1$ which is outside the price region for type 1b and therefore a type 1b equilibrium is also always unique.

Equilibrium 2a: Again, types 1a, 1b and 4 can already be excluded. 2b has a different price range and the same is true for 3b as $s-1$ is greater than β for $s>\beta+\frac{1}{1-\beta}$. For 2a and 3a to happen for the same prices $\frac{1}{1-\beta}p_1-\frac{\beta}{1-\beta}N_1+s-1+\frac{\beta}{1-\beta}\leq(s-\beta)p_1-\beta(s-\beta)N_1+\beta$ has to hold for a feasible p_1. For $s>\beta+\frac{1}{1-\beta}$ the inequality solves to $p_1\geq\beta N_1+1$ which is not a feasible price for 2a and therefore this type is also always unique.

Equilibrium 2b: Here, 3a and 3b both cannot be ruled out immediately so an analysis of the price restrictions for both pairs is necessary. To have both 2b and 3a together for the same price vector $p_1-\beta N_1+s-1-\beta\leq(s-\beta)p_1-\beta(s-\beta)N_1+\beta$ has to hold and this leads to $p_1\geq\beta N_1+\frac{s-2\beta-1}{s-\beta-1}$. For $s>\beta+\frac{1}{1-\beta}$ this is always greater than $\beta N_1+\beta$ and therefore out of the range for equilibrium 2b.

To have both 2b and 3b $p_1-\beta N_1+s-1-\beta\leq p_1-\beta N_1+\beta$ has to hold which leads to $s-2\beta-1\leq 0$ which is impossible for $s>\beta+\frac{1}{1-\beta}$. The 2b equilibrium is therefore also always unique.

Equilibrium 3a: The only non-unique equilibrium left is 3b but this has a different price range than the 3a equilibrium and therefore a type 3a equilibrium is also always unique.

Equilibrium 3b: The type 3b equilibrium has to be also unique as all the others are.

Summary: This shows that the equilibria are indeed unique whenever $s>\beta+\frac{1}{1-\beta}$. The proof further shows that the uniqueness of the consumer stage equilibria are only dependent on the values of s and β but independent of N_1. Therefore, the uniqueness property would not change if one switched the system qualities to for example $s_2=1$ and $s_1=s>\beta+\frac{1}{1-\beta}$.

A.2. Proof: Price Equilibria in Duopoly

A price vector (p_1, p_2) is an equilibrium if one price is optimal given the price of the other firm and vice versa. Optimal means that no other price gives a higher profit (equal to revenue as variable costs are zero) for that firm. However, the situation is complicated due to the fact that a given price vector leads to a consumer equilibrium in the next stage and hence changing the price in this stage might also change the equilibrium type in the next stage. The best way to show the equilibria in this stage is to assume a consumer equilibrium to be played in stage 4 and to find price equilibrium candidates that lead to that consumer equilibrium. For this purpose it is important to note that the combined revenue functions of all the equilibria in the respective price ranges are always continuous. This is intuitive and a proof is given in the next section of this appendix.

Proposition 13. *It is never optimal to choose a price vector that leads to a consumer equilibrium 4 (no system is sold).*

Proof. Such a price vector has to satisfy $p_2 \geq s$ and $p_1 \geq \beta N_1 + 1$ and then the networks and revenues are zero for both firms (see Table 1). However, given the price of the other firm, firm 1 can always set a price to select equilibria 2a or 2b to earn a positive revenue and firm 2 can select equilibria 3a and 3b. Therefore, it is not optimal for either firm to stick to a price such that an equilibrium 4 is played in the next stage. □

Proposition 14. *It is also never optimal to choose a price vector that leads to consumer equilibria 2a and 3a (Only one firm sells system but not the full market is served).*

Proof. For a 2a equilibrium firm 2 has no revenues and $p_2 \geq \frac{1}{1-\beta}p_1 - \frac{\beta}{1-\beta}N_1 + s - 1 - \frac{\beta}{1-\beta} > s - 1 > 0$ has to hold. However, by setting $p_2 < \frac{1}{1-\beta}p_1 - \frac{\beta}{1-\beta}N_1 + s - 1 - \frac{\beta}{1-\beta}$ firm 2 can force a 1a equilibrium to be played for which the revenue for firm 2 is positive because as $p_2 > s - 1$ that critical value is always positive. The situation is the same for the 3a equilibrium for which firm 1 can always set $p_1 < \frac{1}{s-\beta}p_2 + \beta N_1 - \frac{\beta}{s-\beta}$ as $\frac{1}{s-\beta}p_2 + \beta N_1 - \frac{\beta}{s-\beta} > \beta N_1 > 0$. □

Proposition 15. *There is a price equilibrium in stage 3 that leads to a 2b consumer equilibrium in stage 4. In that case,* $p_2 = z_2 = \pi_2 = 0$ *and* $p_1 = \pi_1 = \beta N_1 - s + \beta + 1, z_1 = 1$ *for* $N_1 \geq \frac{2s-3\beta-2}{\beta}$*. No other equilibrium that leads to a 2b consumer equilibrium is possible.*

Proof. As firm 2 does not earn any revenues in this equilibrium the price setting for firm 2 can only be optimal if for any other feasible price p_2 the revenue is non-positive given the price of firm 1. Clearly, a higher price $p_2 > 0$ does not lead to a higher revenue as it still leads to a 2b consumer equilibrium when $p_1 = \beta N_1 - s + \beta + 1$ (Table 1). The price restriction for the 2b equilibrium then turns to $p_2 \geq 0$ and as consumer equilibria are unique the only possible equilibrium to be played for any price p_2 has to be the 2b equilibrium. A lower price $p_2 < 0$ of course is also not possible as prices have to be non-negative. The situation is therefore optimal for firm 2 as it cannot do better by choosing any other price.

The situation is also optimal for firm 1: Given $p_2 = 0$ firm 1 can set a price such that either equilibria 1b, 2b or 3b are played in stage 4. 3b can never be better than 2b for firm 1 as then the revenue is zero. Note also that given a 2b equilibrium is played in stage 4 firm 1 cannot do better by choosing a different price that also leads to a 2b equilibrium: $p_1 = \beta N_1 - s + \beta + 1$ is the maximum feasible price such that a 2b equilibrium will be played and lowering the price would not make sense because the full market is served already. The revenue such that a 1b equilibrium is played is $\pi_1 = \frac{-p_1 + \beta N_1 - \beta}{s - 2\beta - 1}p_1$ and optimizing this quadratic function leads to $p_1 = \frac{\beta N_1 - \beta}{2}$. As the combined revenue function for firm 1 for equilibria 1b and 2b is continuous, playing 2b is only optimal if $\frac{\beta N_1 - \beta}{2} \leq \beta N_1 - s + 1 + \beta$ and this is the case if $N_1 \geq \frac{2s-3\beta-2}{\beta}$. In that case the maximizing price for the 1b equilibrium is outside of it's range and therefore it is optimal to play the border value which is the 2b equilibrium.. Therefore, this price vector and the restriction on N_1 form an equilibrium in stage 3. Note also that $p_1 = \beta N_1 - s + \beta + 1 > 0$ holds when $N_1 \geq \frac{2s-3\beta-2}{\beta}$.

If there was another equilibrium (p_1, p_2) that would lead to a 2b equilibrium in the next stage then $p_2 = p > 0$ would have to hold. However, in that case also $p_1 = p + \beta N_1 - s + \beta + 1 > \beta N_1 - s + \beta + 1$ had to hold to make it an optimal choice for firm 1. In this case firm 2 could set $p_2 < p$ to make a positive revenue because then consumers would play a 1b equilibrium as then $p_2 < p_1 - \beta N_1 - s + \beta + 1$ (see Table 1). This shows that no other price equilibrium leading to a 2b consumer equilibrium is possible. □

Proposition 16. *There is further no price equilibrium that leads to a 3b consumer equilibrium (firm 2 takes over the full market).*

Proof. The situation is similar to the proof above such that the only equilibrium candidate is $p_1 = 0$ and $p_2 = -\beta N_1 + \beta$. In that case firm 2 plays the maximum possible price for the 3b equilibrium (maximizing revenue) and firm 1 cannot choose a different $p_1 > 0$ such that it gets a positive revenue. However, doing a similar analysis as above, the revenue function for the 1b equilibrium for firm 2 is $\pi_2 = \frac{-p_2 - \beta N_1 + s - \beta - 1}{s - 2\beta - 1} p_2$ which maximizes at $p_2 = \frac{-\beta N_1 + s - \beta - 1}{2}$. Due to the same reasoning as above playing a price that leads to this 3b equilibrium is only better than the optimal 1b equilibrium price if $\frac{-\beta N_1 + s - \beta - 1}{2} \leq -\beta N_1 + \beta$ which leads to the condition $N_1 \leq \frac{1 + 3\beta - s}{\beta} < 0$ as $s > \beta + \frac{1}{1-\beta} > 1 + 3\beta$ for $\beta > \frac{1}{2}$. As by assumption $N_1 > 0$ no price equilibrium can exist such that this 3b equilibrium is played in stage 4. □

Proposition 17. *There is also no price vector that leads to a 1a consumer equilibrium (market sharing without serving the full market).*

Proof. The revenue function for firm 1 in this case is $\pi_1 = \frac{p_2 - (s-\beta)p_1 + \beta(s-\beta)N_1 - \beta}{(s-\beta)(1-\beta)-1} p_1$ and the optimal price $p_1 = \frac{1}{2(s-\beta)} p_2 + \frac{1}{2}\beta N_1 - \frac{\beta}{2(s-\beta)}$. Comparing that optimal price to the restrictions from Table 1 however shows the following: $p_1 = \frac{1}{2(s-\beta)} p_2 + \frac{1}{2}\beta N_1 - \frac{\beta}{2(s-\beta)} < \frac{\beta}{s-\beta-1} p_2 + \beta N_1 - \frac{\beta^2}{s-\beta-1}$ leads to $p_2 < \beta + \frac{(s-\beta)(s-\beta-1)}{(s-\beta)(1-2\beta)-1} \beta N_1 < \beta$ for $\beta > \frac{1}{2}$ and $s > \beta + \frac{1}{1-\beta}$. $p_2 < \beta$ however is not possible for any equilibrium type 1a as shown in the price range. Firm 1 would therefore always choose a price outside of the 1a price range as an optimal price, for example the border price between 1a and 1b. □

Proposition 18. *A price vector leading to a 1b consumer equilibrium is possible for* $\underline{N_1} \leq N_1 < \frac{2s-3\beta-2}{\beta}$. *The price vector is* $(p_1, p_2) = (\frac{\beta N_1+s-3\beta-1}{3}, \frac{-\beta N_1+2s-3\beta-2}{3})$ *and this is always an interior solution in the sense that values at the border to 1a are never played. Furthermore,* $\frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)} \leq \underline{N_1} \leq \frac{s-3\beta-1}{2\beta}$ *holds.*

Proof. Using a similar analysis as above one can derive the optimal prices for both firms given the price of the other: $p_1 = \frac{1}{2}p_2 + \frac{1}{2}\beta N_1 - \frac{1}{2}\beta$ and $p_2 = \frac{1}{2}p_1 - \frac{1}{2}\beta N_1 + \frac{s-\beta-1}{2}$. The crossing point of those two functions leads to the price vector shown in the proposition which is an equilibrium candidate as long as it meets the price restrictions. Two more candidates can be derived from the "border function" $p_2 = \frac{s-\beta-1}{\beta}p_1 - (s-\beta-1)N_1 + \beta$ by combining it with the two optimal prices. This price is optimal for the two firms if the maximum of both revenue functions of the 1a and 1b consumer equilibria are right at the border to 1a or even cross it. If this is not the case then either playing a price in 1a or an "interior" price in 1b is optimal for one of the two firms.

Checking the functions for firm 2 shows that this is not the case. To be optimal to play the border function the following inequalities have to hold (from the price restrictions): $p_2^{1a} = \frac{1}{2(1-\beta)}p_1 - \frac{1}{2(1-\beta)}\beta N_1 + \frac{s-\beta s-1}{2(1-\beta)} \geq \frac{s-\beta-1}{\beta}p_1 - (s-\beta-1)N_1 + \beta$ and $p_2^{1b} = \frac{1}{2}p_1 - \frac{1}{2}\beta N_1 + \frac{s-\beta-1}{2} \leq \frac{s-\beta-1}{\beta}p_1 - (s-\beta-1)N_1 + \beta$. This is optimal if for the price of firm 1 $\beta N_1 + \beta\frac{s-3\beta-1}{2s-3\beta-2} \leq p_1 \leq \beta N_1 + \beta\frac{(s-2\beta)(1-\beta)-1}{2(1-\beta)(s-\beta-1)-\beta}$ holds and then $\frac{s-3\beta-1}{2s-3\beta-2} \leq \frac{(s-2\beta)(1-\beta)-1}{2(1-\beta)(s-\beta-1)-\beta}$ is a necessary condition. Solving the inequality however leads to $s \leq 1$ which contradicts the assumption $s > \beta+\frac{1}{1-\beta} > 1$. There is therefore no price p_1 such that a vector (p_1, p_2) where the border price is played is optimal for firm 2. Therefore, an equilibrium always has to be an interior equilibrium.

The other candidate $(p_1, p_2) = (\frac{\beta N_1+s-3\beta-1}{3}, \frac{-\beta N_1+2s-3\beta-2}{3})$ is certainly optimal inside the range of 1b but depending on the actual values N_1 both firms also have the option to choose prices such that another consumer equilibrium is played in stage 4. The analysis is quite complex and I discuss the different situations in 4 steps, splitting the price ranges for both firms:

$0 \leq p_2 \leq \beta \Leftrightarrow \frac{2s-6\beta-2}{\beta} \leq N_1 \leq \frac{2s-3\beta-2}{\beta}$**:**
In this case firm 1 has the alternatives to choose equilibria 2b and 3b instead of 1b. 3b of course can never be better (zero revenue) but 2b could. Using the result from the analysis of the 2b equilibrium from above shows that playing the 1b equilibrium is optimal if $N_1 < \frac{2s-3\beta-2}{\beta}$ which is consistent with $0 \leq p_2 \leq \beta$.

$\beta < p_2 < s-1 \Leftrightarrow \frac{-s-3\beta+1}{\beta} < 0 < N_1 < \frac{2s-6\beta-2}{\beta}$**:**
The alternatives in this case are 1a, 2b and 3a. The zero revenue type 3a cannot be optimal while the 2b type offers less revenues when $N_1 < \frac{2s-3\beta-2}{\beta}$ which is the case here. Furthermore, the 1a consumer equilibrium always offers less revenues than the 1b equilibrium as the optimal price crosses the border to 1b (see proof of inexistence of 1a price equilibrium). Of course, $p_1 \leq \frac{\beta}{s-\beta-1}p_2 + \beta N_1 - \frac{\beta^2}{s-\beta-1}$ has to hold as well as otherwise it is optimal for firm 1 to just play the border price which does not lead to any equilibrium (see above). Using p_1 and p_2 for this inequality leads to $N_1 \geq \frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)} > 0$. Playing this equilibrium candidate is therefore always optimal for firm 1 as long as $\frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)} \leq N_1 < \frac{2s-3\beta-2}{\beta}$ holds.

$0 \leq p_1 \leq \beta N_1 \Leftrightarrow N_1 \geq \frac{s-3\beta-1}{2\beta}$**:**
Here, the alternatives for firm 1 are 2b and 3b which makes it similar to the first analysis above. 2b cannot improve the revenue while 3b potentially can. However, doing a similar analysis to the one above shows that 1b is better than 3b for firm 2 if $N_1 > \frac{-s-3\beta+1}{\beta}$ which is negative, hence, for any $N_1 \geq \frac{s-3\beta-1}{2\beta}$.

$\beta N_1 < p_1 < \beta N_1 + \beta \Leftrightarrow \frac{s-6\beta-1}{2\beta} < N_1 < \frac{s-3\beta-1}{2\beta}$**:**
In this case the alternatives are 1a, 2b, 3a and 3b. 2b (zero revenue) and 3b (see above) can already be ruled out as better options. Furthermore, as was already shown in the analysis above, $N_1 \geq \frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2}> 0$ has to hold because otherwise it does not lead to a 1b equilibrium. As $\frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2} > \frac{s-6\beta-1}{2\beta}$ holds for $s > 1$ the upper bound of the price region $(\beta N_1 + \beta)$ is not an issue. The situation for the combined revenue function is rather complex as 1a, 1b and 3a all have quadratic revenue functions in their regions.

The optimal price for a 3a type equilibrium is $p_2 = \frac{s}{2}$ and the price restriction $p_2 \leq (s-\beta)p_1 - \beta(s-\beta)N_1 + \beta$ has to hold. This is not fulfilled for $p_1 = \frac{\beta N_1 + s - 3\beta - 1}{3}$ and $p_2 = \frac{s}{2}$ as it would need $N_1 \leq \frac{s-3\beta-1}{2\beta} - \frac{3(s-2\beta)}{4\beta(s-\beta)} < \frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)}$. This means that for the N_1 in question and p_1 from the equilibrium candidate firm 2 cannot set it's optimal price for a 3a equilibrium but the best it can do is to set the border price to 1b. Firm 1 can further play the optimal 1a price $p_2 = \frac{1}{2(1-\beta)}p_1 - \frac{1}{2(1-\beta)}\beta N_1 + \frac{s-\beta s-1}{2(1-\beta)}$ in that region or play the border price between 1a and 3a: $p_2 = (s-\beta)p_1 - \beta(s-\beta)N_1 + \beta$. In this situation revenue functions have to be compared directly and playing the equilibrium candidate is optimal for firm 2 if both $\frac{(2s-3\beta-2-\beta N_1)^2}{9(s-3\beta-1)} \geq \frac{(2s-3\beta\frac{s+1}{2}-2-\beta N_1)^2}{(1-\beta)[(s-\beta)(1-\beta)-1]}$ (1b vs. 1a) and $\frac{(2s-3\beta-2-\beta N_1)^2}{9(s-3\beta-1)} \geq \frac{(s-\beta-1)(s-3\beta)+\beta-2\beta(s-\beta)N_1}{3}\frac{2\beta N_1 - s + 3\beta + 4}{3}$ (1b vs. 3a). Unfortunately, the second inequality cannot be solved in generality for N_1[34]. However, the above analysis shows that the lower bound for the region of N_1 where a 1b equilibrium can exist is somewhere in $[\frac{(s-3\beta-1)^2}{\beta(2s-3\beta-2)}, \frac{s-3\beta-1}{2\beta}]$ which defines $\underline{N_1}$. □

[34] The first one leads to $N_1 \geq \frac{2s-3\beta-2}{\beta} - \frac{3(s-1)}{2}\frac{1}{1-\sqrt{\frac{(1-\beta)[(s-\beta)(1-\beta)-1]}{s-2\beta-1}}}$.

A.3. Proof: Continuity of Revenue Functions in Pricing Stage

To show the continuity of the revenue functions one has to show that at the border between two consumer equilibria the revenues are equal. As the prices are the same at the border this is equivalent to showing that the networks are equal. This is clear intuitively when having in mind how the stage 4 equilibria are constructed. Using Table 1 this leads to the following combinations:

1. The 1a equilibrium has a border to the 1b equilibrium which is at $p_2 = \frac{s-\beta-1}{\beta}p_1 - (s-\beta-1)N_1 + \beta$. Plugging it into z_2 (and the equivalent one into z_1) for both equilibrium types 1a and 1b leads to $z_1^{1a} = p_2 - \beta = z_1^{1b}$ and $z_2^{1a} = \frac{-p_1}{\beta} + N_1 + 1 = z_2^{1b}$.

2. The 1a equilibrium also has a border to the 2a equilibrium at $p_2 = \frac{1}{1-\beta}p_1 - \frac{\beta}{1-\beta}N_1 + s - \frac{1}{1-\beta}$ which leads to $z_1^{1a} = -p_2 + s = z_1^{2a}$ and $z_2^{1a} = 0 = z_2^{2a}$.

3. Another border can be found to the 3a equilibrium at $p_2 = (s-\beta)p_1 - \beta(s-\beta)N_1 + \beta$ and then $z_1^{1a} = 0 = z_1^{3a}$ and $z_2^{1a} = -p_1 + \beta N_1 + 1 = z_2^{3a}$.

4. There is also a border between 1b and 2b at $p_2 = p_1 - \beta N_1 + s - \beta - 1$ and one to 3b at $p_2 = p_1 - \beta N_1 + \beta$ and those lead to $z_1^{1b} = 1 = z_1^{2b}$ and $z_2^{1b} = 0 = z_2^{2b}$ and $z_1^{1b} = 0 = z_1^{3b}$ and $z_2^{1b} = 1 = z_2^{3b}$.

Therefore: At each border between two consumer equilibria the network functions are continuous and therefore also the revenue functions.

A.4. Abstract

This master thesis scrutinizes the topic of openness, a generalization of many important subtopics, in information technology and it's economic properties. Examples of subtopics that are included in the definition of openness are open source, compatibility, standards, interfaces and licensing. The work is structured into two major parts: In the first one a study is done about the history and strategy with a focus on openness about Microsoft, Apple and Google, three of the most important companies in information technology. This study shows that those strategies are quite different but also changed over time. While Microsoft adopted a closed approach that opened up a little in recent years (partly because of pressure from competition authorities) Google always relied on openness with the exception of it's most important technology: It's search algorithm. Apple is in many ways even more closed than Microsoft, although it seems to be more open with respect to open source, especially when it improves the quality of it's own products. In the second part of this work an economic model is developed which includes network economics, vertical product differentiation and an incumbent-entrant-scheme with economies of scale. In this 4-stage model with 2 companies it is possible to discuss certain aspects of competition and openness in this industry and this is done with applications of operating systems licensing and some economic aspects and incentives of open source (impossible market entry, signaling incentives, external revenues). The results are dependent on the chosen model parameters but some general results can be shown nevertheless: That licensing is most likely optimal whenever market entry by the second firm is possible and that an open source strategy indeed makes sense when the discussed aspects and incentives hold true.

A.5. Abstract - German

Die vorliegende Diplomarbeit widmet sich dem Thema Offenheit (”openness“) in der Informationstechnik und untersucht dazu ökonomische Zusammenhänge und Gesetzmäßigkeiten. Offenheit ist hier als Verallgemeinerung anderer Themen zu sehen, unter anderem Open Source, Kompatibilität, Standards, Schnittstellen und Lizensierung. Die Arbeit ist dabei in zwei wesentliche Teile gegliedert:

Im ersten Teil geht es um eine Aufarbeitung der Geschichte und Strategie mit besonderem Fokus auf Offenheit von Microsoft, Apple und Google, drei der wichtigsten und erfolgreichsten Unternehmen im Bereich der Informationstechnik. Dabei zeigt sich, dass die Strategien sehr unterschiedlich sind, sich aber auch im Zeitablauf ändern. Während Microsoft traditionell auf Geschlossenheit setzt und sich erst in den letzten Jahren teilweise durch Druck von Wettbewerbsbehörden etwas öffnete (Offenlegung von Schnittstellen, Lizensierung von Technologie, Akzeptanz von Open Source) ist Google ein von Anfang an offenes Unternehmen, welches nur in Ausnahmenfällen geschlossene Systeme verwendet, dies aber dann bei der wichtigsten Technologie, dem Such-Algorithmus. Apple ist in vielen Bereichen noch geschlossener als Microsoft, in anderen wie zum Beispiel aber Open Source durchaus aufgeschlossen wenn es der eigenen Produktqualität nützt.

Im zweiten Teil der Arbeit geht es um die Darstellung der beobachteten Zusammenhänge in einem ökonomischen Modell. Dieses vereint Netzwerkökonomie, vertikale Produktdifferenzierung und ein Incumbent-Entrant-Schema mit fallenden Durchschnittskosten. In diesem 4-Phasen-Modell können verschiedene Aspekte des Themas dargestellt werden, wobei exemplarisch die Lizensierung eines Betriebssystems und gewisse Aspekte von Open Source aus der Literatur (unmöglicher Markteintritt, Signalisierungsanreize, externe Umsätze) beleuchtet werden. Die jeweiligen Resultate hängen letztlich von den gewählten Parametern ab, es zeigt sich aber, dass Lizensierung vor allem dann sinnvoll ist wenn es zu einem Markteintritt kommen würde und dass unter den vorgeschlagenen Annahmen Open-Source für ein potenziell in den Markt eintretendes Unternehmen durchaus Sinn macht.

A.6. Curriculum Vitae

Name:
Markus Wagner BSc.

Born:
March 6th 1984 in Vienna, Austria

Education:
08/2009 - 12/2009: *Austria Illinois Exchange Program, University of Illinois at Urbana-Champaign, USA*
08/2008 - : *Masters Program Computer Technology, Vienna University of Technology, Austria*
03/2005 - : *Diploma Program Economics, University of Vienna, Austria*
10/2004 - 09/2008: *Bachelor Program Electrical Engineering, Vienna University of Technology, Austria*
09/1998 - 06/2003: *Computer Engineering with focus on Environmental Engineering, Higher Technical College Braunau am Inn, Austria*
03/1996 - 06/2004: *Preparatory Course Piano, University of Music and Dramatic Arts Mozarteum Salzburg, Austria*