

EINFÜHRUNGSSTRATEGIEN FÜR DEFACTO STANDARDS IM INTERNET

eingereicht von:

Thomas Aiginger

thomas.aiginger@reflex.at

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Magister rerum socialium oeconomicarumque
Magister der Sozial- und Wirtschaftswissenschaften
(Mag.rer.soc.oec.)

**Fakultät für Wirtschaftswissenschaften und Informatik,
Universität Wien**
**Fakultät für Technische Naturwissenschaften und Informatik,
Technische Universität Wien**

Studienrichtung: Wirtschaftsinformatik

Begutachter:

Univ.-Prof. Dr. Jörg-Martin Pflüger

Wien, im January 2003

*Für meine Eltern,
meinen Bruder Philipp
und Susi*

Ich versichere:

- dass ich die Diplomarbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- dass ich diese Diplomarbeit bisher weder im In- noch im Ausland (einer Beurteilung bzw. einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe.
- dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, im January 2003

Thomas Aiginger

thomas.aiginger@reflex.at

Zusammenfassung

Diese Arbeit analysiert Strategien von Unternehmen zur Einführung von Standards. Der Schwerpunkt liegt auf IT- und insbesondere Internet-Standards, wobei vor allem Standardisierung durch den Markt (De facto-Standardisierung) und weniger formale Standardentwicklung (De jure-Standardisierung) betrachtet wird. Die Arbeit beschreibt Einführungsstrategien anhand der fünf Faktoren Netzwerkeffektivitäten, Free-ridership, Switching Costs, Offenheit und Kooperation. Aus diesen Faktoren wird ein Entscheidungsframework entwickelt, das Unternehmen bei der systematischen Entwicklung einer Einführungsstrategie unterstützen soll.

Inhaltsverzeichnis

I	Einleitung und Definitionen	3
1	Einleitung	4
2	Grundlegende Definitionen	6
3	Vorstellung eines Beispiels	8
II	Theoretische Grundlagen	10
4	Netzwerkexternalitäten und deren Folgen	11
4.1	Netzwerke und Externalitäten	12
4.2	Netzwerkexternalitäten und die ökonomische Theorie	16
4.3	Einführungsstrategien unter dem Gesichtspunkt von Netzwerkexternalitäten	21
5	Standards als Public Goods und die Free-Rider Problematik	24
5.1	Was sind Public Goods?	25
5.2	Einschränken der Nicht-Ausschließbarkeit	26
5.3	Free-Ridership und die Teilnahme am Standard-Entwicklungsprozess	34
5.4	Finanzielle Verwertung von public-good Standards	37
5.5	Konkurrenten ausschließen oder free-rider akzeptieren?	40
6	Switching Costs und Lock-In	43
6.1	Entstehung von Switching Costs	45
6.2	Folgen der Switching Costs für Standard-Strategien	47
7	Offenheit von Standards	51
7.1	Formen der Offenheit	52
7.2	Offenheit an Fallbeispielen	56
8	Kooperation und Kompatibilität	63
8.1	Formen von Kooperationen	64
8.2	Wie Kooperation und Kompatibilität den Wettbewerb verändern	66
8.3	Die Kooperationsentscheidungen mehrerer Unternehmen	69

III	Entscheidungsframework und Ausarbeitung einer Strategie	74
9	Synthese der Grundlagen zu einem Entscheidungsframework	75
9.1	Beschreibung des Frameworks	76
9.2	Phase 1 - Marktanalyse	78
9.3	Phase 2 - Das Grundgerüst der Strategie	79
9.4	Phase 3 - Feinabstimmung der Strategie	83
10	Anwendung des Entscheidungsframeworks auf das fiktive Unternehmen	
	ModernCooking	87
10.1	Phase 1 - Marktanalyse	88
10.2	Phase 2 - Das Grundgerüst der Strategie	93
10.3	Phase 3 - Feinabstimmung der Strategie	98
10.4	Zusammenfassung	103

Teil I

Einleitung und Definitionen

Kapitel 1

Einleitung

Standards¹ spielen eine bedeutende Rolle in der heutigen Ökonomie. Sie ermöglichen uns E-Mails zu versenden, in Geschäften Videos zu kaufen, Strom aus der Steckdose zu beziehen und in der ganzen Welt, ob online oder offline, mit Kreditkarten zu bezahlen. Seit der Automatisierung der industriellen Revolution ist die Bedeutung von Standards stetig gestiegen und in der heutigen Informationstechnologie scheinen sie allgegenwärtig.

Wird oft behauptet, dass im Internet eigene Gesetze gelten, so liegt das vor allem auch daran, dass Standards durch die Natur des Internets eine entscheidende Rolle zukommt. Im Internet und in der Informationstechnologie im Allgemeinen geht es darum Informationen zu verarbeiten, zu speichern und zu kommunizieren und dazu wird ein hohes Maß an Standardisierung benötigt. Egal ob bei Amazon.com ein Buch gekauft oder mit ICQ gechattet wird, werden bei jeder Transaktion über unzählige standardisierte Schnittstellen, auf Ebene des Netzwerks und auf Ebene der Anwendungen, Informationen transportiert. Verwenden die chattenden User oder der Käufer und der Verkäufer unterschiedliche Standards, so wird die Transaktion gar nicht oder nur in eingeschränktem Maße zustande kommen.

Diese Arbeit beschäftigt sich daher mit Strategien, die Firmen anwenden, um ihrem Standard im Internet zur Durchsetzung zu verhelfen. Es geht vor allem darum für Unternehmen, die einen Standard einführen, eine Entscheidungsgrundlage zur Auswahl der richtigen Strategie zu schaffen. Wohlfahrtsökonomische Auswirkungen von Standards oder die Bedeutung von Standards für Konsumenten werden nur dort behandelt, wo sie für die Schaffung dieser Entscheidungsgrundlage wichtig sind.

Das Hauptaugenmerk liegt auf der Einführung von Defacto-Standards, also vom Markt geschaffenen Standards, und nicht auf de jure, von Organisationen wie der ISO, definierten Standards.

Strategien für die Einführung von Standards setzen sich aus verschiedenen miteinander verknüpften Faktoren zusammen, die im Teil II dieser Arbeit dargestellt werden. Der bedeutendste Einflussfaktor auf Standards in Netzwerken wie dem Internet sind Netzwerkexternalitäten, die in Kapitel 4 behandelt werden. Im Kapitel 5 geht es dann um den „public good“-Charakter von Standards und der damit verknüpften Gefahr der „free-ridership“. Ein weiteres Merkmal von Standards sind „switching costs“, die beim Wechsel des Standards auftreten können. Deren Auswirkungen auf Einführungsstrategien werden in Kapitel 6 behandelt. Kapitel 7 beschäftigt sich mit der Frage, in welchen Fällen ein offener und wann ein proprietärer Standard vorzuziehen ist. Im Kapitel 8 werden schließlich Vor- und Nachteile einer Kooperation für die Entwicklung des Standards besprochen.

Teil III diskutiert die Anwendung der Faktoren an einem Beispiel. Zuerst werden im Kapitel 9 Zusammenhänge zwischen den Faktoren aufgezeigt und die Faktoren mit ihren Wechselbeziehungen in ein systematisches Entscheidungsframework zusammengefügt. Dann wird im Kapitel 10 mit Hilfe dieses Entscheidungsframeworks eine konkrete Strategie für die Einführung eines Standards ausgearbeitet.

¹Für eine genaue Definition von Standards siehe Kapitel 2.

Kapitel 2

Grundlegende Definitionen

Folgende Begriffe sind grundlegend für diese Arbeit. Da ihre Definitionen in der Literatur nicht eindeutig sind, soll hier ihre Bedeutung für diese Arbeit definiert werden.

Kompatibilität: Für Kompatibilität bieten Farrell und Saloner[19] eine ausgezeichnete Definition:

„We call products compatible when their design is coordinated in some way, enabling them to work together“

Standards: Ein Standard ist eine explizit oder implizit festgelegte Menge von Regeln, die Kompatibilität oder allgemeiner den Austausch von Informationen ermöglicht. Diese weit reichende Definition umfasst einerseits die große Menge der Kompatibilitätsstandards, die zum Beispiel Verbindungen zwischen unterschiedlichen Systemen oder Systemen und ihren Daten ermöglichen. TCP/IP, der Standard, der es Clients im Internet ermöglicht miteinander zu kommunizieren, ist genauso ein Kompatibilitätsstandard wie die Spezifikation der CD, die garantiert, dass alle CDs auf allen kompatiblen Geräten abgespielt werden können. Doch viele der in dieser Arbeit angesprochenen Konzepte, wie Netzwerkexternalitäten oder „switching costs“, lassen sich nicht nur auf Kompatibilitätsstandards, sondern auf alle Regeln, die den Austausch von Informationen ermöglichen, anwenden. So ist etwa auch ein Online-Diskussionsforum in unserem Sinne ein Standard, da es eine implizit festgelegte Regel ist sich unter einem bestimmten URL zu treffen, um Informationen zu einem bestimmten Thema auszutauschen. Die Definition geht so weit, dass zum Beispiel auch die Sprache als Standard betrachtet werden kann. Und tatsächlich wird man sich für die Einführung einer Sprache (z.B. Esperanto) mit ganz ähnlichen Fragen wie bei der Einführung eines Internet-Standards befassen müssen. Beispielsweise besteht auch hier das Problem, dass niemand bereit ist die Sprache zu erlernen, bevor ihm nicht ausreichend Gesprächspartner zur Verfügung stehen und sich die Sprache nur langsam verbreiten kann (siehe „excess inertia“ in Kapitel 4.2).

Defacto-Standards: Standards, die ihre Akzeptanz durch hohe Verbreitung am Markt erreichen und nicht weil sie von einer formalen Standardisierungsbehörde wie der International Standardization Organisation (ISO) oder der Internet Engineering Task Force (IETF) ausgewählt wurden. Allerdings müssen auch Ergebnisse formaler Standardisierung, auch Dejure-Standards genannt, oft erst durch den Markt akzeptiert werden, sodass sie in der Einführungsphase ähnlichen Effekten wie Defacto-Standards ausgesetzt sind.

Komplementärgüter: Produkte oder Dienstleistungen, die sich gegenseitig bedingen, das heißt erst gemeinsam einen Nutzen oder zumindest gemeinsam einen größeren Nutzen ergeben. So sind zum Beispiel Glühbirne und Lampe oder CDs und CD-Player Komplementärgüter. Damit Komplementärgüter miteinander verwendet werden können, müssen sie kompatibel sein, d.h. dem gleichen Standard entsprechen. So kann eine Glühbirne nur in eine Lampe eingeschraubt werden, wenn beide dem gleichen Standard für die Fassung entsprechen.

Installed Base: Die Menge der Personen oder Unternehmen, die einen Standard verwendet.

Netzwerkexternalitäten: Netzwerkexternalitäten beschreiben den Effekt, dass Standards oft für einen Anwender umso mehr Nutzen erzeugen, je mehr andere Anwender denselben Standard verwenden wie er. In Kapitel 4 wird dieser Begriff und die Unterscheidung zwischen direkten und indirekten Netzwerkexternalitäten noch genauer diskutiert.

Kapitel 3

Vorstellung eines Beispiels

Zunächst soll ein fiktives Beispiel vorgestellt werden, an dem sich die meisten in dieser Arbeit behandelten Aspekte demonstrieren lassen und für das am Ende der Arbeit mit Hilfe des Entscheidungsframeworks eine Strategie entwickelt wird.

In diesem Beispiel geht es darum einen Standard für Kochrezepte im Internet einzuführen. Um die Beschreibung zu erleichtern soll der Standard CookML (in Anlehnung an XML) und die den Standard einführende Firma ModernCooking heißen. ModernCooking produziert internetfähige Küchengeräte, wie zum Beispiel Kühlschränke, die ein Inventar der gekühlten Lebensmittel speichern und Einkaufszettel erstellen oder sogar selbstständig bestellen können. Die Geräte lassen sich über ein zentrales Terminal kontrollieren, das auch eine Datenbank mit Rezepten verwaltet.

Ein wichtiges Ziel des Systems ist es den Kochvorgang zu erleichtern. So kann man sich zum Beispiel eine Liste der mit den vorhandenen Lebensmitteln kochbaren Rezepte anzeigen oder automatisch eine Einkaufsliste der für bestimmte Gerichte fehlenden Zutaten erstellen lassen. Beim Kochen gibt das Terminal schrittweise Anleitungen, kontrolliert die Herdtemperatur, zeigt auf der Waage an, wie viel von welcher Zutat noch hinzugefügt werden soll usw.

Der für diese Arbeit interessante Punkt liegt bei den Rezepten, die über das Internet ausgetauscht und heruntergeladen werden sollen. Damit die Rezepte von den Küchengeräten automatisch verarbeitet werden können, müssen sie einem definierten Standard entsprechen. Der Standard muss zum Beispiel festlegen, welche Kochvorgänge zur Verfügung stehen oder wie Zutaten, Maßeinheiten und Mengenangaben kodiert werden.

Die Firma ModernCooking will nun so einen Standard einführen und möglichst viele Internet-Anwender und Online-Rezeptdatenbanken motivieren ihre Rezepte in diesem Format abzuspeichern. Es ist naheliegend als Sprache für den Standard XML(eXtensible Markup Language) auszuwählen. XML ist eine Metasprache, die es ermöglicht Dokumentformate zu definieren. Neue Formate werden in sogenannten Document Type Definitions (DTD) spezifiziert, die angeben, welche Elemente in welcher Anordnung in Dokumenten verwendet werden dürfen.

Das erste Problem bei der Einführung eines solchen Standards ist, dass solange keine Anwender vorhanden sind auch keine Rezeptdatenbanken diesen Standard adaptieren werden und bevor es keine Rezepte gibt, werden nur wenige Anwender den Standard nützen wollen (Kapitel 4). Weitere Fragen drehen sich um strategische Kooperationen (Kapitel 8) und wie die einführende Firma verhindert, dass sie die Kosten trägt, aber andere Firmen vom Standard genauso profitieren wie sie selbst (Kapitel 5). Ebenso muss sich ModernCooking überlegen, wie offen sie den Standard und seine Entwicklung gestalten wollen (Kapitel 7) bzw. wie sie Anwender möglichst lang an ihren Standard binden können (Kapitel 6).

In der Realität arbeiten Haushaltsgerätehersteller gerade an einem Standard, der die Kommunikation zwischen verschiedenen Haushaltsgeräten ermöglichen soll. Wir setzen für unser Beispiel voraus, dass eine solche Kommunikation auch zwischen Geräten unterschiedlicher Hersteller bereits möglich ist. Wir verwenden für diese Arbeit das fiktive Beispiel des Rezeptstandards, da sich mit ihm Netzwerkeffekte besser demonstrieren lassen als anhand des Kommunikationsstandards. Denn in unserem Beispiel tauschen Anwender untereinander Informationen aus und nutzen nicht nur Haushaltsgerätehersteller den Standard.

Teil II

Theoretische Grundlagen

Kapitel 4

Netzwerkexternalitäten und deren Folgen

Netzwerkexternalitäten sind einer der wichtigsten Faktoren für die Entwicklung und Verbreitung von Standards. Der Begriff der Netzwerkexternalität wurde zum ersten Mal von Rohlfs[43] erwähnt, gewann aber erst durch die Arbeit von Katz und Shapiro 1985 [26] größere Bedeutung. Er beschreibt, dass in vielen Fällen große Netzwerke für Konsumenten attraktiver als kleine sind. Gibt es zum Beispiel zwei konkurrierende Standards für Kochrezepte im Internet so würden Konsumenten den Standard bevorzugen, den schon mehr Anwender einsetzen. Denn durch die Größe des Netzwerks steigt die Anzahl der Personen, mit denen Rezepte getauscht werden können, und die Anzahl der Firmen, die Produkte (Software, Küchengeräte) für diesen Standard entwickeln.

Zunächst werden wir Netzwerkexternalitäten präziser definieren (Kapitel 4.1), dann werden deren Folgen mit Hilfe verschiedener ökonomischer Theorien präsentiert (Kapitel 4.2). Im Kapitel 4.3 werden dann die Auswirkungen von Netzwerkexternalitäten auf Einführungsstrategien von Standards näher beleuchtet.

4.1 Netzwerke und Externalitäten

Betrachten wir zuerst den ersten Teil der Netzwerkexternalität: das Netzwerk. Netzwerkexternalitäten können sowohl in realen Netzwerken, wie zum Beispiel Telefon- oder Eisenbahnnetzwerken, als auch in virtuellen Netzwerken, wie zum Beispiel dem Netzwerk der Anwender des VHS-Standards, auftreten.

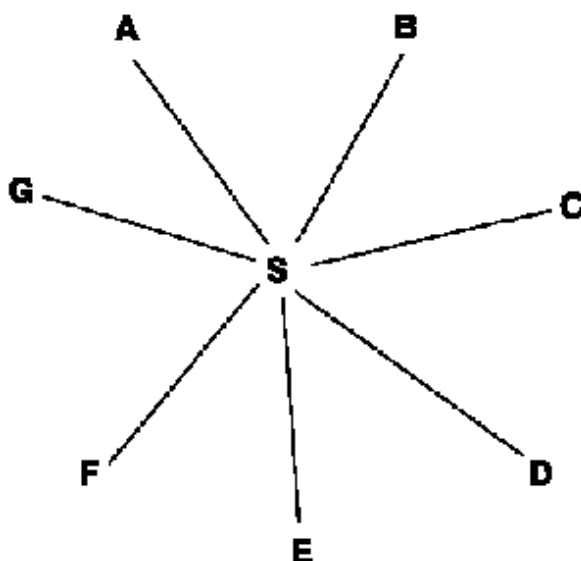


Abbildung 4.1: Beispiel eines two-way Netzwerks. Quelle [11]

Economides [11] unterscheidet bei der Analyse von Netzwerkexternalitäten zwischen one-way und two-way Netzwerken. Abbildung 4.1 zeigt das klassische Beispiel eines two-way Netzwerks. Die Darstellung kann zum Beispiel als Telefonnetzwerk interpretiert werden, bei dem eine Person vom Anschluss A über den Switch S mit einer Person am Anschluss B telefoniert. Obwohl die Verbindungen AS und BS sehr ähnlich sind, handelt es sich nicht um Substitute sondern um Komplementärgüter. Denn erst die Kombination von zwei der Verbindungen AS, BS, CS, DS, usw. erzeugt einen Nutzen für den Anwender, eine einzelne Verbindung hat keinen stand-alone

value. Ein weiteres wichtiges Merkmal von two-way Netzwerken ist, dass sowohl die Verbindung ASB als auch BSA existiert und die beiden unterscheidbar sind. Es kann sowohl eine Person A eine Person B anrufen als auch umgekehrt und es macht, zum Beispiel bei der Bezahlung, einen Unterschied wer wen anruft. Außerdem werden in two-way Netzwerken Anwender meist mit den Netzwerkknoten identifiziert. Der Anschluss A ist eindeutig einem Kunden zugeordnet. Andere Beispiele für two-way Netzwerke sind zum Beispiel Straßennetze oder peer-to-peer Anwendungen im Internet (z.B. Napster, ICQ...)

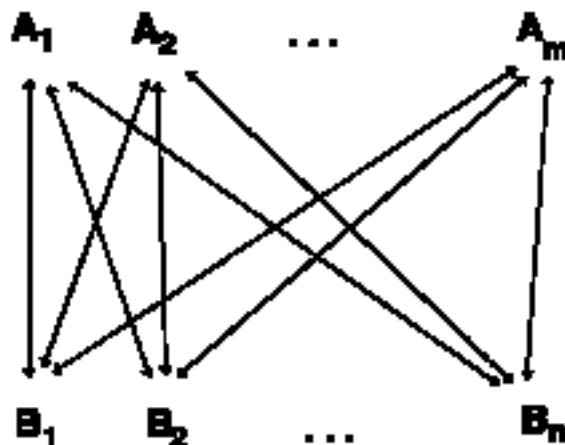


Abbildung 4.2: Beispiel eines one-way Netzwerks. Quelle [11]

Von einem one-way Netzwerk spricht man hingegen, wenn eine der beiden Verbindungen BA oder AB nicht existiert, ökonomisch nicht sinnvoll ist oder wenn die Verbindung AB identisch mit BA ist, weil im Netzwerk keine Richtungen unterschieden werden können. In typischen one-way Netzwerken existieren zwei unterschiedliche Arten von Komponenten und ein nutzbares Gut kann nur durch die Kombination zweier Elemente, eines jeder Art, erzeugt werden. Abbildung 4.2 zeigt ein one-way Netzwerk, bei dem die Knoten $A_1..A_m$ zum Beispiel als Bankomaten interpretiert werden können und die Knoten $B_1..B_n$ als Bankinstitute in diesem Bankomatennetzwerk, von denen Geld behoben werden kann. Eine Geldbehebung von einem Konto auf der Bank B_5 vom Bankomaten A_1 ist die einzig mögliche Verbindung A_1-B_5 und ist identisch mit der Verbindung B_5-A_1 . Auch hier sind die Komponenten A_1 und B_5 Komplementärgüter, die nur gemeinsam einen Nutzen für den Anwender ergeben.

One-way Netzwerke sind formal identisch mit vertikal verbundenen Märkten¹[11]. Abbildung 4.2 könnte genauso als Darstellung des CD-Player und des CD-Tonträger Marktes interpretiert werden. Erst durch eine Kombination A_1-B_5 eines CD-Players A_1 mit einer CD B_5 wird für den Anwender ein Nutzen erzeugt. Die Verbindungen A_1-B_5 und B_5-A_1 sind wieder identisch. Äquivalente Beispiele vertikal verbundener Märkte sind Computerhardware und kompatible Betriebssysteme oder Betriebssysteme und dazu kompatible Anwendungen.

In dem Beispiel des Rezeptstandards spielen nun sowohl one-way als auch two-way Netzwerke eine Rolle.

¹Vertikal verbundene Märkte sind Märkte, die auf der Wertschöpfungskette hintereinander liegen, z.B. der Markt für Videorecorder und der Markt für VHS-Kassetten, im Gegensatz zu horizontal verbundenen Märkten, wie dem Markt für DVD-Player und dem Markt für VHS-Videorecorder.

Erstens besteht ein two-way Netzwerk aus Anwendern des Standards, die untereinander Rezepte tauschen. Ein Knoten kann als Anwender, der Rezepte im CookML Standard verwendet oder als Rezeptdatenbank eines Anwenders interpretiert werden. Wie bei den meisten two-way Netzwerken stehen die Knoten damit in direkter Verbindung mit einem Anwender. Eine Übertragung eines Rezeptes von B zu A unterscheidet sich von einer Übertragung von A zu B. Ein Knoten allein erzeugt im Kontext des Rezeptaustausches keinen Nutzen, zum Tausch wird eine Verbindung zweier Knoten benötigt, es handelt sich also um Komplementärgüter.

Zweitens besteht eine vertikale Verbindung zwischen Küchengeräten, die Rezepte interpretieren, und den Rezepten selbst. Auf diese Verbindung treffen die Eigenschaften von one-way Netzwerken zu. Es gibt nur eine mögliche Verbindung zwischen einem Rezept und einem bestimmten Küchengerät und, was für unsere Analyse der Netzwerkexternalitäten entscheidend sein wird, weder Rezepte noch Küchengeräte lassen sich direkt einem Anwender zuordnen. Denn mehrere Anwender können das gleiche Rezept verwenden, während die Rezeptdatenbank eines Anwenders per definitionem eben diesem Anwender zugeordnet ist. Bei den Küchengeräten spielt es keine Rolle, dass jedes Gerät einem Anwender gehört, denn es geht um die Anzahl unterschiedlicher Arten von Geräten und nicht um die Zahl der produzierten Geräte einer Art. So könnte zum Beispiel A_1 ein Kühlschrank von Miele, A_2 ein Kühlschrank von Siemens und A_3 eine Waage sein, aber nicht A_1 die Waage der Familie Huber und A_2 die Waage der Familie Maier.

Nun wenden wir uns dem zweiten Teil des Wortes Netzwerkexternalität zu. Externalitäten treten auf, wenn ein Marktteilnehmer einen anderen beeinflusst, ohne dass Kompensationen geleistet werden^[48]. Ein klassisches Beispiel für Externalitäten wäre die Verschmutzung eines Flusses durch eine Industrieanlage, die die Einnahmen eines Fischereibetriebes am gleichen Fluss senkt. Der Industriebetrieb beeinflusst den Fischer (negativ), ohne dafür Kompensationen zu bezahlen. Von Netzwerkexternalitäten spricht man dann, wenn ein Produkt für einen Konsumenten umso mehr Nutzen ergibt, je mehr andere Konsumenten das gleiche Produkt verwenden, also je größer das Produktnetzwerk ist. Während die Verschmutzung negative Externalitäten erzeugt, sind Netzwerkexternalitäten immer positiv. Ein Anwender, der ein Produkt erwirbt oder einen Standard adaptiert, vergrößert das Netzwerk und erzeugt damit Netzwerkexternalitäten für die bestehenden Anwender.

Je nach Art des Netzwerks können unterschiedliche Arten von Netzwerkexternalitäten unterschieden werden. Wenn Konsumenten mit den Gütern identifiziert werden können, wie in two-way Netzwerken, treten direkte Netzwerkexternalitäten auf. In einem Telefonnetzwerk mit n Teilnehmern gibt es $n \times (n - 1)$ potentielle zusammengesetzte Güter (Anrufe). Jeder neue Kunde, der dem Netzwerk beiträgt, verursacht direkte Netzwerkexternalitäten für alle anderen Anwender, indem er dem Netzwerk $2 \times n$ neue potentielle Güter hinzufügt².

In one-way Netzwerken mit m Komponenten $A_1..A_m$ und n dazu kompatiblen Komponenten $B_1..B_n$, gibt es $m \times n$ potentielle zusammengesetzte Güter. Ein neuer Anwender im Netzwerk erzeugt nun nur indirekte Netzwerkexternalitäten indem er die Nachfrage nach Gütern A und/oder B erhöht. Je mehr Anwender CD-Player (Komponente A) besitzen, umso größer wird die Nachfrage nach CDs (Komponente B) sein. Aufgrund von steigenden Skalenerträgen³ stellen CD-Produzenten umso mehr CDs her, je höher die Nachfrage ist.

Tritt ein neuer Anwender dem Netzwerk des Rezeptstandards bei, indem er sich ein Küchengerät kauft oder seine Rezepte im CookML Format abspeichert, so hat dies zwei Effekte. Erstens

²Er ermöglicht n neue Verbindungen, deren Richtungen unterscheidbar sind, daher $2 \times n$

³Steigende Skalenerträge (engl.: economies of scale) bedeuten, dass der Ertrag für eine Einheit des Gutes steigt, je höher die Anzahl der verkauften Güter ist.

schafft er für die bereits bestehenden Anwender des Standards direkte Netzwerkexternalitäten, da er ihnen zusätzliche Tauschmöglichkeiten eröffnet. Zweitens erzeugt er indirekte Netzwerkexternalitäten, indem er die Nachfrage nach weiteren Küchengeräten, nach Anwendungen für den Standard und nach Rezepten erhöht.

Der entscheidende Grund für das Entstehen von Netzwerkexternalitäten ist aber immer die Komplementarität zweier Güter, was sich an Hand des Rezeptstandards schön zeigen lässt. Nehmen wir an es ist nicht möglich fremde Rezepte downzuloaden oder sich Rezepte auf CD-ROM zu kaufen, sondern es können nur die eigenen Rezepte in Küchengeräte eingegeben werden. So würden keine direkten Netzwerkexternalitäten entstehen, denn es gäbe keine zweite komplementäre Rezeptdatenbank, mit der ein Austausch möglich wird. Dennoch würden indirekte Netzwerkexternalitäten dadurch entstehen, dass die Rezepte komplementär zu den Küchengeräten sind. Denn die Rezeptdatenbank eines Anwenders im CookML-Format erhöht die Nachfrage dieses Anwenders nach weiteren CookML Küchengeräten. Analog könnte auch eine Situation ohne indirekten Netzwerkexternalitäten konstruiert werden, indem die Rezeptdatenbank mit einem bestimmten Küchengerät verbunden wird und nicht für weitere Geräte verwendet werden kann.

Zusätzlich zu diesen direkten und indirekten Netzwerkexternalitäten identifiziert Mertz[34] noch die folgenden Vorteile eines größeren Netzwerks für den Anwender. Diese Vorteile werden in der Literatur als Netzwerkeffekte bezeichnet, sind im Allgemeinen schwächer als Netzwerkexternalitäten und treten auch in traditionellen Märkten ohne Komplementaritätsbeziehungen auf:

- Bei länger nutzbaren Gütern, wie z.B. Mobiltelefonen, führt ein größeres Netzwerk auch zu einem größeren **second-hand Markt**
- Für stärker verbreitete Produkte werden mehr **Support**, Beratung nach dem Kauf oder Reparaturdienstleistungen angeboten. Beispielsweise gibt es eine große Anzahl von Consultants, die auf die marktführende betriebswirtschaftliche Software SAP R/3 spezialisiert sind.
- Je mehr Anwender eine Technologie benutzen, umso mehr **Produktinformationen** stehen zur Verfügung, z.B. in Internet-Newsgroups.
- Wird für ein Produkt eine **spezifische Ausbildung** benötigt, wie beispielsweise für eine Programmiersprache, so stehen für weiter verbreitete Produkte mehr ausgebildete Fachkräfte zur Verfügung.

Netzwerkexternalitäten führen dazu, dass schon erfolgreiche Firmen noch erfolgreicher werden, da Anwender das größte Netzwerk bevorzugen. Neue Anwender adaptieren damit die erfolgreichste Technologie, vergrößern das Netzwerk noch mehr und es kommt zu einer positiven Feedback-Schleife. Ähnliche Feedback-Effekte sind schon länger bekannt und treten auch in traditionellen Märkten in Form der schon oben erwähnten steigenden Skalenerträge auf. Gerade in der IT-Branche spielen diese, auf Grund von niedrigen variablen Kosten, eine wichtige Rolle. Wenn zum Beispiel ein Betriebssystem einmal entwickelt ist, können neue Kunden mit nur geringen zusätzlichen Kosten bedient werden. Das Unternehmen mit den meisten Kunden kann die Fixkosten auf die größte Anzahl von Käufern verteilen, damit am kostengünstigsten anbieten und zieht so noch mehr Kunden an. Dieser Effekt wird auch als angebotsseitige Skalenerträge bezeichnet. Netzwerkexternalitäten sind dem gegenüber nachfrageseitige Skaleneffekte, da eine grosse Absatzmenge nicht eine *angebotsseitige* Kostenersparnis bringt, sondern die Kundennachfrage erhöht.

Ein wichtiger Unterschied ist, dass angebotsseitige Skaleneffekte meist bei Mengen unterhalb des Marktvolumens erschöpft sind. Höhere Produktionsmengen führen zum Beispiel in der Automobilindustrie nur bis zu einem bestimmten Limit zu Effizienzsteigerungen. Ab diesem Limit nehmen die Schwierigkeiten eine grosse Organisation zu managen überhand und senken die Effizienz[48]. Märkte mit nachfrageseitigen Skaleneffekten sind hingegen oft „winner-takes-all“ Märkte, die zu Monopolen führen. Denn bei Netzwerkexternalitäten gibt es kein Limit, ab dem sie sich zu negativen Effekten verkehren. Auch wenn schon 99% der Anwender ihre Rezepte im CookML Standard speichern, so erhöhen sich die Netzwerkexternalitäten trotzdem, wenn das letzte Prozent dem Netzwerk auch noch beiträgt.

4.2 Netzwerkexternalitäten und die ökonomische Theorie

In diesem Kapitel sollen die wichtigsten Implikationen von Netzwerkexternalitäten anhand verschiedener ökonomischer Theorien aufgezeigt und anschließend an zwei Beispielen aus dem Bereich elektronischer Bezahlung veranschaulicht werden.

Die meisten Theorien unterscheiden zwischen zwei Faktoren. Erstens dem „stand-alone value“ einer Technologie, also dem Nutzen, den sie auch dann bietet, wenn kein Netzwerk vorhanden ist. Zweitens existiert ein Netzwerkfaktor, der eine Funktion der Größe des Netzwerks ist.

Farrell und Saloner [13] modellieren eine Situation, in der es eine alte (U) und eine neue (V) konkurrierende Technologie gibt. Ein User, der vor der Entscheidung steht welche Technologie er adaptieren soll, wählt zwischen dem Nutzen

$$u(x) = a + bx$$

für die Adaption der Technologie U und dem Nutzen

$$v(y) = c + dy$$

für Adaption der Technologie V, wobei x und y die Netzwerkgrößen, a und c die stand-alone values und bx und by die netzwerkabhängigen Nutzen der beiden Technologien sind.

Damit die neue Technologie adaptiert wird, muss nun entweder der stand-alone value c der neuen Technologie groß genug sein, um den schon aufgebauten Netzwerkvorteil der alten Technologie auszugleichen, oder es wird eine „**kritische Masse**“ an Anwendern benötigt um der neuen Technologie zur Durchsetzung zu verhelfen. Die kritische Masse für die Technologie U ist die Netzwerkgröße \bar{x} , sodass gilt:

$$\bar{x} = \frac{c - a + d \times y}{b}$$

Liegt die Anzahl der Anwender unter dieser kritischen Masse \bar{x} , so wird die Technologie nicht adaptiert, da der Nachteil des kleineren Netzwerks den Vorteil des möglicherweise höheren stand-alone values dominiert.

Im idealen Fall setzt sich der Standard durch, der den Gesamtnutzen⁴ aller User maximiert. Doch Farrell und Saloner identifizieren auch zwei mögliche nicht wohlfahrtsoptimale Ergebnisse des Modells: „excess inertia“ und „excess momentum“.

Im Falle von excess inertia wird die wohlfahrtsoptimale Technologie nicht adaptiert, weil es dem Unternehmen nicht gelungen ist eine ausreichende kritische Masse aufzubauen. Oder anders ausgedrückt, kein Anwender möchte das Risiko tragen als Erster ein neues Produkt zu adaptieren, ohne zu wissen, ob das Netzwerk jemals eine ausreichende Größe erreichen wird.

Verfügen die Anwender über perfekte Information,⁵ so können sie das Auftreten von excess inertia verhindern. Farrell und Saloner [19] erklären das mit folgender Rückwärtsinduktion:

⁴Die Summe aller Nutzenwerte der einzelnen Anwender.

⁵Perfekte Information bedeutet in diesem Fall, dass jeder Anwender die Präferenzen der anderen Anwender kennt und daher ihre Entscheidung antizipieren kann.

Haben bereits alle bis auf einen Anwender zum neuen Standard gewechselt, so wird der Letzte auf Grund des hohen Netzwerkvorteils ebenfalls wechseln. Bei perfekter Information weiß der vorletzte Anwender, dass der Letzte wechseln wird und wechselt ebenfalls. Genauso antizipiert der drittletzte die Entscheidung der beiden letzten, usw. Imperfekte Information, wie sie in der Realität auftritt, unterbricht diese Kette. Der erste Anwender kann nicht vorhersehen, wie sich alle anderen Anwender entscheiden und geht daher nicht das Risiko ein den neuen Standard zu adaptieren. Selbst wenn alle Anwender den neuen Standard bevorzugen, kann excess inertia auftreten, weil die Anwender keine Informationen über das Verhalten der anderen haben. Farrell und Saloner zeigen auch, dass sich diese Koordinationsproblematik durch Kommunikation lösen lässt, wenn alle Anwender die gleichen Präferenzen haben⁶. Dieses Ergebnis ist allerdings für die Einführung eines Internetstandards mit tausenden Anwendern weniger relevant, da bei großen, dezentral organisierten Gruppen eine solche Kommunikation eben nicht möglich ist.

Doch auch das Gegenteil von excess inertia kann auftreten. Excess momentum beschreibt den Zustand der nicht wohlfahrtsoptimalen Standardisierung. Das kann einerseits dadurch eintreten, dass ein neuer Standard auf Grund seines hohen stand-alone values adaptiert wird, aber der Verlust der Netzwerkexternalitäten für die bestehende installed base der alten Technologie, den Wohlfahrtsgewinn der neuen User übertrifft. Hätten die neuen User die alte Technologie adaptiert, so wäre es zwar für sie selbst nicht optimal gewesen, doch der Gesamtnutzen aller User wäre durch die auftretenden Netzwerkexternalitäten maximiert worden[34]. Die zweite Möglichkeit für excess momentum liegt dann vor, wenn die falsche Technologie zu schnell adaptiert wird, so dass später in den Markt eintreffende Konsumenten auf Grund des hohen Netzwerkvorteils keine andere Wahl haben als auch die schlechtere Technologie zu adaptieren. Grund dafür sind „first-mover advantages“, wie sie weiter unten noch genauer beschrieben werden.

Wie auch schon in der Rückwärtsinduktion zur Lösung des excess inertia Problems erkennbar ist, spielen die Erwartungen der Anwender eine große Rolle. Farrell und Saloner [13] untersuchen deshalb in ihrem Modell die Auswirkungen von Produktvorankündigungen. Sie nehmen an, dass Unternehmen das Erscheinen eines neuen Produktes ankündigen können, bevor es tatsächlich erhältlich ist. Die Strategie der Vorankündigung kann das Ergebnis der Adaption entscheidend beeinflussen, da der Netzwerkvorteil des Konkurrenten zum Zeitpunkt der Vorankündigung noch kleiner ist als später bei der tatsächlichen Einführung. Außerdem werden Anwender, die im Zeitraum zwischen der Vorankündigung und der Produkteinführung die Konkurrenztechnologie erworben hätten, dazu bewogen ihre Entscheidung bis nach den Einführungstermin zu verschieben.

Die Bedeutung von Konsumentenerwartungen wird auch dadurch unterstrichen, dass zum Beispiel im Modell von Katz und Shapiro[27] statt des tatsächlichen Netzwerknutzens der vom Konsument erwartete Nutzen des Netzwerks modelliert wird. Wie bei den Vorankündigungen wird auch hier die Adaption eines Standards entscheidend von Marketingmaßnahmen und dem Image des Standards beeinflusst.

Im Modell von Katz und Shapiro wird nicht der stand-alone value, sondern der Preis der Technologie als netzwerkunabhängiger Faktor verwendet. Katz und Shapiro zeigen in ihrem Modell, dass Firmen ihre Technologie in der Einführungsphase sponsern, d.h. unter Kosten verkaufen, um einen Netzwerkvorteil zu erlangen. Voraussetzung für sponsorship ist, dass Firmen Eigentumsrechte für diese Technologien besitzen oder andere Marktzutrittsbarrieren⁷ existieren, damit verhindert werden kann, dass die sponsorship der Konkurrenz zu Gute kommt. Äquivalent

⁶Haben Anwender unterschiedliche Präferenzen, so erhöht Kommunikation die excess inertia, weil Anwender nicht ihre wahren Präferenzen offenbaren.

⁷Verschiedene Marktzutrittsbarrieren werden im Kapitel 5.2 näher behandelt.

zu excess inertia und excess momentum zeigen auch Katz und Shapiro eine Möglichkeit einer ineffizienten Nichtstandardisierung und eine Möglichkeit ineffizienter Standardisierung auf. Ersteres wird damit erklärt, dass Anwender nur ihren eigenen Nutzen maximieren, nicht aber die Netzwerkexternalitäten beachten, die sie für andere erzeugen.

Zu einer ineffizienten Standardisierung, also der Adaption eines unterlegenen Standards, kann es nach Katz und Shapiro durch einen „first-mover advantage“ kommen. In diesem Fall kann sich die Technologie, die in der ersten Periode überlegen erscheint, einen so großen Netzwerkvorteil sichern, dass sich Konsumenten der zweiten Periode trotz einer überlegenen anderen Technologie ebenfalls für den Sieger der ersten Periode entscheiden. Ein solcher ineffizienter first-mover advantage kann dadurch ausgelöst werden, dass ein Unternehmen seine Technologie früher auf den Markt bringt, durch Marketingmaßnahmen zu Beginn verschleiert wird, welche Technologie besser ist, oder indem eine Technologie in der ersten Periode gesponsert wird.

Die wichtigsten Ergebnisse dieser Theorien sollen noch einmal kurz zusammengefasst werden:

- Der Nutzen eines Netzwerkproduktes ist abhängig von dem stand-alone value des Produkts und der Netzwerkgröße.
- Erreicht ein Standard nicht die kritische Masse, so bleibt sein Netzwerk zu klein um sich durchsetzen zu können.
- Der wohlfahrtsoptimale Standard wird nicht immer adaptiert, da sich Anwender nicht koordinieren können (excess inertia).
- Aufgrund eines first-mover advantages wird manchmal ein nicht wohlfahrtsoptimaler Standard eingeführt. Die Standardisierung erfolgt zu früh und kann auf Grund des hohen Netzwerkvorteils nicht mehr rückgängig gemacht werden (excess momentum).
- Die Erwartungen von Konsumenten über die zukünftige Netzwerkgröße des Standards spielen eine entscheidende Rolle.
- Durch Vorankündigungen können Unternehmen die Einführungschancen ihres Standards verbessern.
- Unternehmen, die Marktzutrittsbarrieren (z.B. Eigentumsrechte) für ihren Standard kontrollieren, sponsern ihn in der Einführungsphase, d.h. bieten den Standard unter Kosten an.

Anhand dieser Modellergebnisse soll nun eine erfolgreiche Einführung eines Standards mit einer gescheiterten verglichen werden. Als Beispiel dienen zwei Standards für elektronische Bezahlung, das erfolgreiche PayPal-System und eines der zahlreichen gescheiterten electronic cash Verfahren: eCash von DigiCash Inc.

Prinzipiell haben beide Systeme das gleiche Anwendungsgebiet, sie können erstens für den kommerziellen B2C⁸ Bereich verwendet werden, etwa um im Internet einzukaufen. In diesem Bereich konkurrieren sie allerdings mit den mächtigen und bereits etablierten Kreditkartengesellschaften. Zumindest während der Anfangsphase erscheint daher ihre zweite Anwendung, der C2C Bereich, erfolversprechender. In diesem Bereich gab es zum Beispiel für die Bezahlung von in Internet-Auktionen ersteigerten Waren keine verbreitete Lösung.

Beim System von DigiCash werden elektronische Münzen von kooperierenden Banken ausgegeben. Anwender müssen auf ihrem PC eine Software installieren, eine so genannte „purse“, die

⁸B2C steht für „Business-to-Consumer“ und damit für Geschäftsbeziehungen zwischen Konsumenten und Unternehmen. Im Gegensatz dazu steht C2C für Transaktionen zwischen Konsumenten.

die elektronischen Münzen speichert. Bei der Bezahlung werden die Münzen an den Empfänger übermittelt, der bei seiner Bank die Gültigkeit der Münzen überprüfen und die Münzen gegen Geld einlösen kann. Durch raffinierte, von DigiCash patentierte kryptographische Verfahren konnte sowohl hohe Sicherheit als auch hohe Anonymität gewährleistet werden.

Mit dem PayPal-System werden Zahlungen quasi per E-Mail versendet. Ein registrierter User hat ein Konto bei PayPal, auf das er mit Kreditkarte oder Überweisung Geld einzahlen kann. Auf der PayPal Website kann er dann einen Betrag und eine E-Mail Adresse angeben, an die er diesen Betrag versenden will. Der Empfänger erhält eine E-Mail mit einem Hyperlink, das die Überweisung identifiziert. Ist der Empfänger schon PayPal-Mitglied so kann er durch Anklicken des Links den Betrag auf sein PayPal-Konto überweisen. Ist er noch kein Mitglied, kann er in wenigen Schritten ein PayPal Konto eröffnen, auf das er den Betrag einzahlt. Das Geld vom Konto kann er sich entweder mit Hilfe eines Schecks ausbezahlen lassen oder selbst für weitere Transaktionen verwenden. Zu Beginn wurde PayPal nur durch Zinsgewinne des auf PayPal-Konten liegenden Geldes finanziert, mittlerweile hebt PayPal von professionellen Anwendern 3% Transaktionsgebühr ein. Bezüglich Sicherheit und Anonymität kann das PayPal-System nicht mit eCash mithalten. PayPal kämpft mit zahlreichen Betrugsfällen, sowohl von Käufer- als auch von Verkäuferseite[40] [5], und mit Klagen von Usern deren Konten gesperrt wurden[57].

Heute hat PayPal nach eigenen Angaben 12 Mio. Kunden, es werden täglich 20.000 neue Konten eröffnet und seit April 2002 schreibt das Unternehmen schwarze Zahlen[18]. DigiCash ist 1998 in Konkurs gegangen, ohne dass eCash jemals eine nennenswerte Useranzahl gewonnen hätte.

Wo liegt der Unterschied zwischen den beiden Technologien? Warum hat die technologisch überlegene Technologie verloren?

Potentielle Kunden für eCash waren alle Kunden von kooperierenden Banken. Denn hatte ein Anwender kein Konto bei einer partizipierenden Bank, so gab es für ihn keine Möglichkeit eCash Münzen zu erwerben. Natürlich war es für DigiCash nicht leicht neue Banken zu gewinnen, da kaum Händler eCash akzeptierten, bevor eine nennenswerte Anwenderschicht vorhanden war. Es handelt sich um das altbekannte Problem mit dem Huhn und dem Ei oder ökonomisch ausgedrückt um excess inertia. Keine Händler ohne Anwender, keine Anwender ohne Bank und keine Bank ohne Händler. DigiCash ist nie über Pilotversuche mit Banken hinausgekommen.

PayPals potentielle Kunden sind hingegen alle Anwender, die E-Mail verwenden und entweder eine Bankverbindung oder eine Kreditkarte besitzen. Hier tritt kein „Huhn-Ei“ Problem auf, da auch Personen, die nicht Mitglied bei PayPal sind, Zahlungen per E-Mail empfangen und sich mit wenigen Schritten registrieren können. eCash User oder Händler mussten dafür erstens ein Konto bei der richtigen Bank haben und zweitens im Vergleich zu PayPal relativ umständliche Software installieren.

Analysieren wir diese Überlegung genauer in der Sprache unseres Modells. Erstens können wir feststellen, dass User beider Standards signifikante direkte Netzwerkexternalitäten erzeugen. Erst mit einem zweiten Anwender gemeinsam kann eine Transaktion zustande kommen. Jeder Anwender, der dem Netzwerk beiträgt, erhöht damit die Anzahl der Transaktionsmöglichkeiten.

Der entscheidende Vorteil von PayPal, dass auch an Nicht-Mitglieder Zahlungen verschickt werden können, kann nun auf zwei Arten im Modell abgebildet werden:

1. Dieser Vorteil erhöht den stand-alone value a des Systems, da PayPal auch genutzt werden kann, wenn noch keine anderen Anwender bei PayPal registriert sind. Die Netzwerkgröße x steht in diesem Fall für die Anzahl der tatsächlich registrierten User.

oder

2. Wir inkludieren alle User an die prinzipiell eine PayPal Zahlung geschickt werden kann, also alle User mit E-Mail Adresse und Bankverbindung, in der Netzwerkgröße x . PayPal hätte also nur einen geringen stand-alone value, wäre dafür aber gleich mit beachtlichen Netzwerkvorteilen gestartet.

Die erste Variante erscheint in diesem Fall praktikabler, da hier auch die Größe des Netzwerks der tatsächlich registrierten User einfließt, die zum Beispiel wesentlich ist um Unternehmen zur Akzeptanz von PayPal zu bewegen.

Für einen unregistrierten Anwender, der soeben eine Bezahlung über PayPal erhalten hat, ist der stand-alone value des PayPal Standards gleich der Höhe der Überweisung. Damit hat der User bereits eine beträchtliche Motivation dem Netzwerk beizutreten, unabhängig davon, wie viele andere User bereits zum PayPal Netzwerk gehören.

Weiters wurde jeder neue PayPal Kunde in der Anfangsphase mit einem Guthaben von 10\$ ausgestattet. (Nachdem sich ein Netzwerk aufgebaut hatte, wurde die Prämie für Neulinge reduziert.) Für diese Kunden betrug der stand-alone value des PayPal Systems 10\$. Hätte Digi-Cash eine ähnliche Promotionsaktion gestartet, so wäre der stand-alone value für ihre Kunden trotzdem unter 10\$ gelegen, da im Falle einer kleinen Netzwerkgröße x , noch keine Transaktionspartner vorhanden waren. Die Wahrscheinlichkeit, dass ein Kunde diese 10\$ für sich sinnvoll verwenden hätte können, wäre geringer gewesen als beim PayPal System, wo im Prinzip jeder andere Internetuser als Transaktionspartner in Frage kam.

Der höhere Installationsaufwand, den eCash-Kunden tragen mussten, minderte den stand-alone value des Standards weiter. eCash hatte, abgesehen von etwaigen Promotionsangeboten, also einen negativen stand-alone value. Ist noch kein Netzwerk vorhanden, so investiert der neue Anwender seine Zeit in die Installation, erhält dafür aber keinen Nutzen.

Der Vorteil von eCash wäre in seiner überlegenen Technologie gelegen, die anonyme und sichere Transaktionen ermöglicht. Dieser Vorteil spiegelt sich im Koeffizienten b des Netzwerkvorteils $b \times x$ wieder. Es ist durchaus anzunehmen, dass das b für eCash größer als für PayPal war. Allerdings ist dieser Faktor nie zum Tragen gekommen, da der zweite Faktor des Produkts, die Netzwerkgröße x , nie einen signifikanten Wert erreicht hat.

In der Vorgehensweise von PayPal können wir auch die Strategie der sponsorship erkennen. Zuerst wurde in die Netzwerkgröße investiert, es wurden keine Gebühren verlangt und neue User erhielten sogar eine Gutschrift von 10\$. Nachdem das Netzwerk die kritische Masse erreicht hatte, wurden die Gutschriften reduziert und Gebühren eingeführt.

PayPals Erfolg in einem Markt, in dem schon unzählige Firmen ihr Glück versucht haben, aber fast alle in Konkurs gegangen sind, wurde also durch die folgenden Maßnahmen erreicht:

1. PayPal wurde auf Kosten von Sicherheit und Anonymität so einfach wie möglich gehalten, sodass Anwender dem Netzwerk mit wenigen Schritten beitreten konnten (geringe „switching costs“).
2. Zu Beginn wurde in das Wachstum des Netzwerks investiert.
3. Durch die minimalen Voraussetzungen werden keine Anwender von vornherein ausgeschlossen (große Anzahl potentieller Kunden)
4. Anwender werben in einer Art Schneeballeffekt selbstständig neue Anwender, indem sie Personen mit PayPal bezahlen, die selbst noch keine registrierten PayPal-User sind.

4.3 Einführungsstrategien unter dem Gesichtspunkt von Netzwerkexternalitäten

Zum Abschluss dieses Kapitels werden die Konsequenzen von Netzwerkexternalitäten zu Einführungsstrategien von Standards und den weiteren Themen dieser Arbeit in Beziehung gesetzt.

Die entscheidende Frage der meisten Standardeinführungen wird sein, wie excess inertia überwunden werden kann. Die einfache Gleichung des Anwendernutzens lautet, wie oben dargestellt:

$$u(x) = a + bx$$

Es gibt nun zwei extreme Strategien, um für einen Anwender Nutzen zu schaffen. Entweder der neue Standard besticht durch einen hohen stand-alone value a oder der Standard kann bereits mit großem Netzwerk x gestartet werden, indem zum Beispiel ein vorhandenes Netzwerk genutzt wird. Shapiro und Varian [48] definieren zwei diesem Ansatz ähnliche Strategien, die sie als „Revolutions-“ und „Evolutions“-Strategie bezeichnen. Erstere kommt zur Anwendung, wenn eine Technologie eingeführt wird, die Konkurrenztechnologien so hoch überlegen ist, dass auf Netzwerkvorteile verzichtet werden kann. Zweitere wird für weniger überlegene Produkte eingesetzt und nutzt ein vorhandenes Netzwerk oder ermöglicht Usern einer bestehenden Technologie einen einfachen Übergang durch Kompatibilität. Shapiro und Varian sprechen von einem trade-off zwischen den beiden Strategien, da Kompatibilität zu einem bestehenden Netzwerk die Performance des neuen Standards beeinträchtigen kann. Entlang dieses trade-offs sind auch Mischformen zwischen den beiden polaren Strategien möglich.

Bei der **Revolutionsstrategie** erfolgt der Aufbau des Netzwerks weitgehend ohne Netzwerkexternalitäten, der stand-alone value des Standards ist hoch genug, dass ihn Anwender auch bei kleinem Netzwerk adaptieren. Bei manchen Standards sind Netzwerkexternalitäten auch nicht von Natur aus vorhanden, sondern werden erst gezielt eingesetzt, wenn der Standard bereits eine ausreichende Verbreitung erreicht hat. Bei einem Online-Buchshop, wie zum Beispiel amazon.com, treten Netzwerkexternalitäten nicht zwingend auf. Ein Kunde kann genauso gut Bücher kaufen, wenn er nur einer von 100 Kunden ist. Es treten lediglich traditionelle Skaleneffekte, wie Bekanntheit, höhere Effizienz bei größeren Absatzmengen usw. auf. Doch wenn der Onlineshop einmal ein ausreichendes Netzwerk aufgebaut hat, so kann dieser Netzwerkvorteil gegenüber Konkurrenten eingesetzt werden. Zum Beispiel indem eine recommendation engine⁹ eingeführt wird, deren Nutzen eindeutig von der Anzahl der einbezogenen User abhängt.

Ein weiteres Element der Revolutions-Strategie ist, zu Beginn User mit hohen Qualitätsansprüchen anzusprechen. Denn für Anwender, die bessere Performance zu schätzen wissen, ist der stand-alone value einer verbesserten Technologie noch höher. Die CD wurde zum Beispiel zu Beginn der 80er Jahre zuerst im klassischen Musiksegment eingeführt, da Hörer klassischer Musik den Vorteil der besseren Klangqualität besonders hoch bewerten. Dadurch war der stand-alone value für sie soviel größer als der von LPs, dass sie ein zu Beginn kleineres Netzwerk in Kauf nahmen.

Eine Variante dieser Strategie lässt sich anwenden, wenn nicht jeder Anwender gleich viel Netzwerkexternalitäten erzeugt. Bei einem MP3-Sharing System, wie zum Beispiel Napster, erhöht ein User, der dem Netzwerk 1000 Lieder beisteuert, den Nutzen 100 mal mehr, als ein User mit nur 10 Liedern. Eine sinnvolle Strategie wäre es, solch wichtigen Usern besondere

⁹Eine recommendation engine wertet die Einkäufe aller User aus und entwickelt aus diesen Informationen Vorschläge, welche Artikel einen Kunden interessieren könnten. Wenn viele User, die sich ein „Psycho“-Video gekauft haben, auch Stephen King Bücher kaufen, dann erhält ein Kunde, der gerade „Psycho“ gekauft hat, eine Leseempfehlung für Stephen King.

Funktionen zur Verfügung zu stellen, die sie auch ohne großes Netzwerk nutzen können, also für sie den stand-alone value zu erhöhen, zum Beispiel durch eine große Menge Speicherplatz auf dem Server oder einer integrierten Datenbank zur komfortablen Verwaltung großer Dateibestände.

Eine Möglichkeit, eine **Evolutionsstrategie** anzuwenden, ist ein bereits vorhandenes Netzwerk zur Verbreitung des Standards zu verwenden. So nützt zum Beispiel die e-payment Technologie Paybox das bestehende Handynetzwerk für die Abwicklung von Zahlungen. Anwender müssen sich zwar noch immer bei Paybox registrieren, um das Service zu nutzen, die „switching costs“ (siehe auch Kapitel 6) werden aber durch Nutzung eines bereits vorhandenen Netzwerks reduziert. Das ist der wesentliche Grundgedanke, der hinter der Evolutionsstrategie steht: den Wechsel zur neuen Technologie, eben die „switching costs“, für den Anwender so einfach wie möglich zu gestalten, um rasch ein Netzwerk aufzubauen. Anders ausgedrückt, mindern Kosten, die User für die Einführung eines Standards aufbringen müssen (Installationsaufwand...), den stand-alone value des Standards. Während bei der Revolutions-Strategie der stand-alone value durch überlegene Performance erhöht wird, wird er bei der Evolutionsstrategie durch Senken der „switching costs“ verbessert. Eine zweite Variante der Evolutionsstrategie liegt darin den Standard kompatibel zu bestehenden Standards zu gestalten. Das hat Borland versucht, indem ihre Quattro Pro Software Makros des Marktführers Lotus 1-2-3 interpretieren konnte und eine ähnliche Menüführung verwendete. Microsoft inkludierte in den ersten Microsoft Word Versionen eine eigene Hilfe, um Word Perfect Usern den Umstieg zu erleichtern. [48]

Im vorigen Kapitel wurde das Ansprechen einer großen potentiellen Kundenschicht als wichtiger Grund für PayPals Erfolg identifiziert. Prinzipiell ermöglicht eine größere Anzahl an erreichbaren Kunden auch ein schnelleres Wachstum des Netzwerks. Allerdings muss ein größeres Kundenpotential nicht immer einen entscheidenden Vorteil darstellen. PayPal wurde in der ersten Ausbaustufe vor allem dazu verwendet, mit eBay gekaufte Güter zu bezahlen. Güter von eBay werden eher lokal als über große Entfernungen verkauft, da es oft schwierig und teuer ist sie weltweit zu versenden. Ein amerikanischer PayPal-User erzeugt demnach für einen europäischen PayPal-Anwender weniger Netzwerkexternalitäten als für einen zweiten amerikanischen PayPal-User. Netzwerkexternalitäten können also auch einen lokalen Charakter haben oder auf andere Faktoren wie Alters-, Geschlechts- oder Nationalitätengruppen beschränkt sein. Deshalb kann ein Vergrößern der Reichweite, zum Beispiel vom nordamerikanischen auf den weltweiten Markt, manchmal nicht die erwünschten Steigerung des Netzwerknutzens mit sich bringen.

Das Gegenstück zur excess inertia ist der first-mover advantage. Der first-mover advantage erhöht die strategische Bedeutung des richtigen Einführungszeitpunktes. Denn derjenige, der als erster beginnen kann sein Netzwerk aufzubauen, hat einen entscheidenden Vorteil, der oft nicht wieder aufgeholt werden kann. In den Erfolgsgeschichten von Amazon, PayPal und eBay spielt es immer eine entscheidende Rolle, dass diese Firmen als erste am Markt waren, die Nachahmer sind selten erfolgreich. Dennoch müssen Unternehmen einen trade-off zwischen frühest möglichem Einsatzzeitpunkt und ausgereifter Technologie finden und vermeiden Konsumenten mit einer überhasteten, fehlerhaften Einführung für immer zu verlieren. Die oben diskutierte Strategie der Vorankündigung kann verwendet werden, um den first-mover advantage „künstlich“ herzustellen oder den Vorteil einer schneller eingeführten Konkurrenztechnologie zu verringern.

Auch sponsorship wird eingesetzt um den first-mover advantage zu erhöhen, damit der Zeitraum, in dem ein Standard als Einziger am Markt ist, besonders gut genützt werden kann. Steht für Konsumenten bereits eine Konkurrenztechnologie zur Wahl, so ist sponsorship schon weit weniger effektiv, da auch die sponsorship der Konkurrenz ausgeglichen werden muss. Der zweite Grund sponsorship anzuwenden ist den stand-alone value der Technologie zu erhöhen, solange noch kein ausreichendes Netzwerk aufgebaut ist, und damit excess inertia zu verhindern. Auf

jeden Fall sollte sponsorship sehr gezielt eingesetzt werden um ein bestimmtes Resultat zu erreichen, zum Beispiel eine bestimmte attraktive Usergruppe für sich zu gewinnen. Besonders vorteilhaft ist es Investitionen zu tätigen, die sich selbst verstärken, wie die 10\$ Prämie von PayPal, die viele geworbene Anwender dazu verwendeten einen weiteren neuen Anwender zu bezahlen und damit anzuwerben.

Katz und Shapiro haben gezeigt, dass Konsumenten einen Standard nicht auf Grund seiner tatsächlichen, sondern seiner erwarteten Netzwerkgröße wählen. Das erhöht die Bedeutung von Marketingmaßnahmen, da zu den üblichen positiven Einflüssen von Marketing auch noch höhere erwartete Netzwerkexternalitäten hinzukommen. Auch die weiteren Themen dieser Arbeit haben Auswirkungen auf die Konsumentenerwartungen. Eine Strategie der Offenheit, im Gegensatz zu einer proprietären Strategie, erhöht die Zuversicht der Konsumenten, dass sie einerseits kein Lock-In in diesem Standard zu befürchten haben und dass der Standard andererseits auch von anderen Unternehmen unterstützt werden kann(siehe Kapitel 7). Ebenso unterstützt eine Kooperation mit starken und bekannten Partnern die Zuversicht von Konsumenten, dass der Standard Chancen hat sich durchzusetzen (siehe Kapitel 8).

Das Konzept der Netzwerkexternalitäten wird uns durch die ganze Arbeit begleiten und bei allen Entscheidungen eine wichtige Rolle spielen. Die nächsten Kapitel werden behandeln, ob es sinnvoll ist einen Standard als „public good“ zu gestalten, welche Formen der Kooperation eingegangen werden sollen und ob es besser ist einen offenen oder einen proprietären Standard zu wählen. Hinter jeder dieser Fragen steht letztendlich ein trade-off zwischen größerem Netzwerk und höheren Gewinnraten.

Kapitel 5

Standards als Public Goods und die Free-Rider Problematik

„Public goods“ sind Güter, die jedem frei zur Verfügung stehen. Wie wir sehen werden begünstigt das zwar eine schnelle Verbreitung, doch stellt sich die Frage, wer Kosten für frei verfügbare Güter trägt. In diesem Kapitel wird gezeigt, dass Standards in vielen Fällen die Eigenschaften von public goods haben und welche Konsequenzen das auf Einführungsstrategien von Standards hat.

Im Kapitel 5.1 werden die Begriffe „public good“ und „free-ridership“ definiert. Dann gilt es im Bezug auf Standards vier Fragen zu klären:

1. Wie kann verhindert werden, dass ein Standard zu einem public good wird?
2. Wenn ein Standard ein public good ist, kann sich eine Kooperation für die Entwicklungsarbeit bilden oder wählen alle Beteiligten die Möglichkeit der free-ridership?
3. Wie kann ein Unternehmen einen public good Standard in finanziellen Gewinn umsetzen?
4. Wann ist es vorteilhaft einen Standard mit public good Charakter einzuführen, in welchen Fällen ist es besser free-ridership zu verhindern?

Die Fragen 1 und 4 legen die Grundlage für die Fragen 2 und 3. Wenn verhindert werden kann und verhindert werden soll, dass ein Standard public good wird, dann stellen sich die Fragen 2 und 3 nicht mehr. Andernfalls gibt es zwei Aspekte zu betrachten: Die Gefahr der free-ridership vor Entwicklung und Verbreitung des Standards (Frage 2) und nach der Einführung (Frage 3).

5.1 Was sind Public Goods?

Güter werden als „public goods“ bezeichnet, wenn die folgenden beiden Eigenschaften auf sie zutreffen[9]:

1. Nicht-Rivalität: Die Konsumation des Gutes durch eine Person verhindert nicht die Konsumation durch andere Personen.
2. Nicht-Ausschließbarkeit: Niemand kann an der Konsumation des Gutes gehindert werden.

Ein Beispiel für ein klassisches public good ist ein Leuchtturm. Jedes Schiff, das entlang der Küste segelt, kann seinen Dienst in Anspruch nehmen, ohne dass (1.) die Konsumation der Dienstleistung durch ein Schiff die Konsumation durch ein anderes Schiff verhindert und dass (2.) Schiffe davon ausgeschlossen werden können die Dienstleistung zu konsumieren, also das Warnlicht des Leuchtturms zu sehen.

Die Eigenschaft der Nicht-Ausschließbarkeit führt zu dem free-rider Problem. Ein free-rider ist ein Konsument oder ein Produzent, der von Handlungen anderer profitiert, ohne dafür zu bezahlen. Dadurch, dass niemand von einem public good ausgeschlossen werden kann, ist es schwierig dafür Geld einzuheben, da es ohnehin jeder gratis beziehen kann. Dadurch kann es zu einer Unterversorgung mit public goods kommen.

Es gibt verschiedene Ansätze zur Lösung des free-rider Problems:

1. Sozialer Druck führt dazu, dass Personen oder Firmen „freiwillig“ an der Erstellung eines public goods teilnehmen. Diese Lösung funktioniert am besten in kleinen Gruppen. z.B.: Bezahlung eines Wachdienstes für eine Nachbarschaft.

2. Ein Beitrag zum public good wird rechtlich vorgeschrieben. Etwa indem das public good durch Steuereinnahmen oder Gelder einer Organisation mit Pflichtmitgliedschaft (z.B.: Kammer) bezahlt wird.
3. Einschränken der Nicht-Ausschließbarkeit. Zum Beispiel könnten free-rider von der Nutzung möglichst eng verbundener Komplementärgüter ausgeschlossen werden. Im Falle eines Leuchtturms könnte das etwa eine Art Leuchtturmgebühr für Schiffe, die den angrenzenden Hafen verwenden, sein. Es kann zwar jedes Schiff das Licht des Leuchtturms verwenden, aber von der Verwendung des eng verbundenen Komplementärgutes des angrenzenden Hafens kann es ausgeschlossen werden.

Nach Kindleberger [28] haben viele Standards ebenfalls die Eigenschaften der Nicht-Rivalität und der Nicht-Ausschließbarkeit. Kindleberger nennt als Beispiel Standards für Maßeinheiten, Bezahlung und Sicherheit, aber auch IT-Kompatibilitätsstandards sind public goods.

Die Spezifikation des Rezeptstandards in Form einer XML-DTD kann (1.) von beliebig vielen Personen oder Firmen verwendet werden, ohne den Nutzen für weitere Personen zu senken. Im Gegenteil, je mehr Personen den Rezeptstandard verwenden, umso höher ist der Nutzen für den Einzelnen. (2.) ist es unmöglich einzelne Personen davon abzuhalten sich an die XML-DTD zu halten.

Dementsprechend ist auch die free-rider Problematik für Standards relevant. Niemand hat Interesse daran die Kosten für Entwicklung und Einführung eines Standards zu tragen, wenn er selbst auch bei Nicht-Beteiligung nicht von der Nutzung des Standards ausgeschlossen werden kann und dem Aufwand der Einführung kein anderer Nutzen, wie zum Beispiel Macht über den Standard, gegenüber steht. Dadurch kann es nach der ökonomischen Theorie zu einer Unterversorgung mit Standards kommen.

Die drei oben angeführten Ansätze zur Lösung des Problems können auch verwendet werden, um eine Unterversorgung mit Standards zu verhindern. Wird der Standard nur innerhalb einer kleinen Gruppe benötigt, so kann diese durch interne Koordination free-ridership verhindern. Die meisten offiziellen Organisationen zur Entwicklung von formellen Standards (z.B.: DIN, NIST, ANSI) werden staatlich gefördert. Vor allem Qualitäts- und Sicherheitsstandards, aber zum Beispiel auch zahlreiche Internet-Kompatibilitätsstandards, wurden auf staatliche Initiative entwickelt. Diese beiden Ansätze helfen einer Firma, die vor der Einführung eines Internet-Standards steht, relativ wenig, sofern sie nicht zufällig ohnehin staatliche Unterstützung bekommt. Deshalb beschäftigt sich das nächste Kapitel mit der dritten Möglichkeit, durch welche zusätzlichen Maßnahmen die Nicht-Ausschließbarkeit eingeschränkt werden kann.

Auch wenn Konkurrenten nicht von der Nutzung des Standards ausgeschlossen werden können, ergeben sich für das Unternehmen, das einen Standard einführt, oft zusätzliche Wettbewerbsvorteile zum Beispiel durch Einflussmöglichkeiten auf die Entwicklung des Standards. Solche Vorteile werden in den Kapiteln 5.3 und 5.4 beschrieben.

5.2 Einschränken der Nicht-Ausschließbarkeit

Es ist weder gewünscht noch möglich zu verhindern, dass ein Standard Eigenschaft (1.) der Nicht-Rivalität von public goods entspricht. Kooperationen, die Einführungskosten von Standards tragen, stört nicht per se, dass Standards public goods sind, sondern dass free-rider den gleichen Nutzen des Standards genießen, ohne Kosten dafür zu tragen. Dafür ist aber Eigenschaft (2.) die Nicht-Ausschließbarkeit verantwortlich.

Allerdings ist es in den meisten Fällen gewünscht den Standard zumindest auf einer Ebene nicht ausschließbar zu gestalten, da der Standard möglichst große Verbreitung finden soll. Um die verschiedenen Möglichkeiten der Ausschließbarkeit analysieren zu können soll zunächst ein kleines Modell zur graphischen Beschreibung der verschiedenen Ebenen von Standards eingeführt werden.

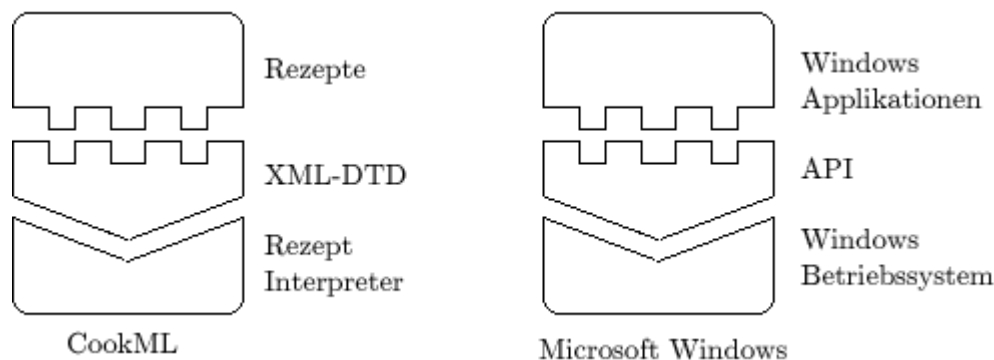


Abbildung 5.1: Schichtenmodell des Rezeptstandards und des Windows API

Abbildung 5.1 zeigt eine schematische Struktur des Rezeptstandards und des Windows Application Programming Interfaces (API). Die mittlere Ebene beschreibt jeweils den Standard. Sie entspricht der XML-DTD der Rezepte bzw. der Definition der möglichen Windows-API Aufrufe. Diese mittlere Ebene ist das, womit sich diese Arbeit beschäftigt, das was es einzuführen und zu definieren gilt.

Die darunter liegende Ebene stellt die Interpretation des Standards dar. Im Fall des Windows-API werden die dem Standard gehorchenden Befehle durch das Windows Betriebssystem verarbeitet. Bei Rezepten implementiert zum Beispiel ein Interpreter in Küchengeräten diese unterste Ebene.

Die oberste Ebene ist das Objekt, das dem Standard gehorcht. So gehorchen Anwendungsprogramme, die am Windows-Betriebssystem laufen, dem Windows-API und Rezepte, die von den Küchengeräten verarbeitet werden, der XML-DTD des Rezeptes.

Bezüglich der free-rider Frage ergibt sich nun eine neue detailliertere Sicht. In den beiden Fällen der Abbildung 5.1 ist es das Ziel der Einführung des Standards, dass auf der obersten Ebene „gefreeridet“ werden kann. Der Standard soll also ein public good sein, denn die Firmen, die den Standard eingeführt haben, erzielen ihren Gewinn auf der unteren Ebene.¹ So erwünscht die public-good Eigenschaften auf der oberen Ebene sind, so unerwünscht sind sie auf der unteren Ebene. Der Hersteller von Küchengeräten, die Rezepte interpretieren, hat zumindest kein direktes Interesse daran, dass auch jede andere Küchengeräte-Firma den Standard kostenlos verwenden kann, ohne an den Einführungskosten des Standards beteiligt gewesen zu sein.² Auch für Microsoft wäre es schädlich, wenn ein anderer Anbieter ein Betriebssystem entwickelt, das den Windows-API Standard interpretieren kann. Im Fall von Microsoft ist die Lage sogar noch eindeutiger, da Microsoft durch seine marktbeherrschende Stellung keine indirekte Hilfe

¹Bei Microsoft liegt der Fall nicht ganz so klar, da Microsoft auf der obersten Ebene Komplementär-Produkte anbietet, z.B.: Ms Office. Für dieses Kapitel, die Analyse der primären Strategie Microsofts eine möglichst hohe Verbreitung des API-Standards zu erreichen, ist dieser Punkt aber nicht relevant.

²Mögliches indirektes Interesse besteht darin, dass er höhere Konkurrenz als trade-off für höhere Verbreitung des Standards akzeptieren kann. Siehe Kapitel 5.5

bei der Verbreitung seines Betriebssystem-Standards benötigt und durch einen Konkurrenten die Monopolstellung gefährdet wäre.

Das Modell hat gezeigt, dass der Aufbau der Standardhierarchie bei beiden Beispielen äquivalent ist.

Warum erscheint es trotzdem intuitiv weniger wahrscheinlich, dass ein Konkurrent die Windows-API für free-ridership ausnützt als einen Rezeptstandard? Die marktbeherrschende Stellung Microsofts würde sicher keinen Konkurrenten davon abhalten auf der untersten Ebene freezuriden. Im Gegenteil, die hohe Verbreitung von Windows-Programmen und die verbreitete Abneigung gegenüber Microsoft würden die Motivation eher verstärken. Der wesentlichste Unterschied, im Bezug auf mögliche free-ridership, zwischen den beiden Beispielen liegt in der Komplexität der untersten Ebene. Ein alternatives Betriebssystem zu entwickeln ist ungleich mehr Aufwand als eine einfache XML-DTD in Küchengeräten zu interpretieren. Dieses Argument wird noch verstärkt, wenn man bedenkt, dass die Weiterentwicklung des Standards in den Händen Microsofts bzw. ModernCookings liegt. Veränderungen am Windows API müssen nun von der Konkurrenz nachgezogen werden, was auf Grund der Komplexität des Windows API einen enormen Aufwand und damit zeitlichen Vorsprung für Microsoft bedeuten kann. Eine weitere Folge der Komplexität ist, dass eine Alternativimplementation performance- und ablaufsicherheitsmäßig hinter dem Original-Windows hinterherhinken wird, da die Windows-Logik übernommen werden muss, aber Implementation und Algorithmen des Original-Windows geheim sind.

Dennoch gibt es einige Bestrebungen weitere untere Layer für das Windows-API, Konkurrenzprodukte zu Microsoft Windows, zur Verfügung zu stellen. Das von namhaften Unternehmen wie Corel unterstützte WINE-Projekt entwickelt zum Beispiel eine Windows API-Implementation für Unix[56]. Der durchschlagende Erfolg ist bisher ausgeblieben, größtenteils wegen der oben angeführten Gründe.[32]

Damit haben wir die erste Möglichkeit herausgearbeitet die Nicht-Ausschließbarkeit einzuschränken: Ausschluss durch hohe Komplexität.

Als nächstes soll die Möglichkeit des Ausschlusses durch rechtliche Maßnahmen, wie Patente, Copyright oder Lizenzen betrachtet werden. Zunächst werden in dem folgenden Exkurs über den „Schutz geistigen Eigentums und Standardisierung“ die Grundlagen dafür beschrieben.

Exkurs: Schutz geistigen Eigentums und Standardisierung

Das Thema, ob die Form des Patent- und Copyrightschutzes für die Anforderungen moderner Informationstechnologie angebracht ist, ist derzeit Gegenstand zahlreicher Diskussionen, von Newsgroups bis zu akademischen Papers. Die Sachlage erfordert eine Fall-zu-Fall Betrachtung, ist komplex und bis dato juristisch nicht eindeutig geregelt. Shapiro und Varian [48, S. 286] schreiben über den Lotus-Borland Prozess (siehe unten)

This case highlights the presence of legal uncertainty over what degree of imitation is permissible; the courts are still working out the limits on how patents and copyrights can be used in network industries

Dieser Exkurs wird daher zunächst den Sollzustand des Urheberrechts in Netzwerkmärkten aus wohlfahrtsökonomischer Sicht behandeln. Es sei darauf hingewiesen, dass es lediglich eine Beschreibung optimaler Bedingungen, nicht aber der realen gegenwärtigen Rechtslage ist, denn deren Diskussion würde eine eigene Arbeit beanspruchen. Zum Abschluss wird aber an einigen

Gerichtsurteilen exemplarisch gezeigt, dass die ökonomischen Überlegungen in der gegenwärtigen Rechtsprechung nur teilweise umgesetzt werden.

Aufgabe des Copyrights und Patentrechts ist es nicht per se geleistete Arbeit zu belohnen, sondern die gesellschaftliche Wohlfahrt zu optimieren, indem Innovationen und die Veröffentlichung dieser Innovationen geschützt wird. Hätten Innovatoren keinen Schutz für ihre Erfindungen, wäre die Motivation in Innovation zu investieren und vor allem erzielte Ergebnisse zu veröffentlichen sehr gering, es würde zu einer Unterversorgung mit Innovationen kommen.

Farrell [14] argumentiert, dass für Märkte in denen Netzwerkexternalitäten vorherrschen, ein geringerer Schutz des geistigen Eigentums wohlfahrtsoptimal wäre. Erstens werden unter der Annahme, dass für geschützte Produkte höhere Preise verlangt werden können, weniger Konsumenten ein geschütztes, teureres Gut kaufen. Treten aber Netzwerkexternalitäten auf, so schadet jeder Konsument, der von dem Netzwerk durch zu hohe Preise abgehalten wird, den das Produkt adaptierenden Konsumenten.

Zweitens existiert in Netzwerkmärkten meist ein starker first-mover advantage (siehe Kapitel 4). Das Produkt, das als Erstes am Markt war, hat einen deutlichen Vorteil und wird sich oft sogar als einziger Defacto-Standard durchsetzen. Durch rechtlichen Schutz für eine gewisse Zeit wird dieser ohnehin schon mächtige Vorteil noch verstärkt, weil der Aufbau von Konkurrenz in der Anfangsphase verhindert wird. Durch den first-mover advantage kommt es schon ohne rechtlichen Schutz zur Bildung suboptimaler Defacto-Standards, warum soll diese Gefahr noch verstärkt werden?

Drittens können durch starken Schutz geistigen Eigentums in Netzwerkmärkten weitere Innovationen verhindert werden. In traditionellen Märkten wird eine schwache Innovation bald durch eine stärkere Innovation der Konkurrenz ersetzt, noch bevor die gesetzliche Schutzfrist abgelaufen ist. Die aufgebauten Netzwerkexternalitäten erschweren aber bei fehlender Kompatibilität den Wechsel zu neueren, besseren Technologien. Verhindert gesetzlicher Schutz Kompatibilität mit der bestehenden Technologie, kann das Innovationen hemmen.

Viertens sollen Interface Spezifikationen nicht durch Copyright schützbar sein [53]. Wie oben beschrieben, tendieren Netzwerkindustrien zur Bildung von Defacto-Standards. Lässt sich nun das Interface (zu Hardware, Software oder User) eines Defacto-Standards durch Copyright schützen, so kommt es zu einer Ausdehnung des Monopoles auf die Komplementärgüter des Standards. Wäre zum Beispiel das Rezeptformat CookML der Defacto-Standard für Rezepte im Internet und die DTD durch Copyright von ModernCooking geschützt, so würde das ein Monopol für Rezept-interpretierende Küchengeräte von ModernCooking bedeuten. Dieser Punkt ist allerdings aus wohlfahrtsökonomischer Sicht weniger eindeutig, da durch Schutz der Interfaces zum Beispiel auch free-ridership verhindert werden kann. Aus ökonomischer Sicht ist daher eine Entscheidung auf Fall-zu-Fall Basis angebracht.

Zusammenfassend kann man also sagen, dass aus Wohlfahrtssicht der rechtliche Schutz für Produkte mit Netzwerkexternalitäten gering gehalten werden sollte. Nun werden wir die Umsetzung dieser Prinzipien in der Praxis anhand dreier Entscheidungen verfolgen: Lotus gegen Borland, Sega gegen Accolade und Amazons Patentierung des one-click purchasing.

Borland implementierte in Quattro Pro seinem Konkurrenzprodukt zu der damals marktführenden Lotus 1-2-3 Tabellenverarbeitung eine Funktion, um 1-2-3 Makros abzuspielen. Lotus klagte daraufhin auf Copyright Verletzung. In erster Instanz wurde zu Gunsten von Lotus entschieden, die Entscheidung wurde aber in der zweiten Instanz revidiert, mit der Begründung, dass durch Inkompatibilität der Makro-Schnittstelle, eher von Lotus zu Borland wechselnde User

bestraft würden, die ihre Investitionen in 1-2-3 Makros durch den Wechsel verlieren. Diese Entscheidung entspricht unserer ökonomischen Analyse und spricht gegen den Copyright-Schutz für Schnittstellen.

Eine ähnliche Entscheidung wurde im Fall Sega gegen Accolade getroffen. Accolade hat Spiele für Segas Spielkonsolen hergestellt. Der Interface Standard für die Spiele war geheim, so dass nur Sega kompatible Spiele erzeugen konnte. Accolade fand diesen Standard allerdings durch Analyse der Konsolenhardware und Reverse Engineering heraus. Das Gericht entschied gegen Segas Klage und befand das Reverse Engineering für legal, da es keinen anderen Weg gab an die Schnittstellenspezifikation heranzukommen. Wenn Firmen durch vertragliche Vereinbarungen verhindern, dass komplementäre Produkte anderer Hersteller verwendet werden, also zum Beispiel Spiele der Konkurrenz auf einer Sega-Spielkonsole, so führt das zu kartellrechtlichen Konsequenzen. Es ist daher nur konsequent, dass Verhindern komplementärer Konkurrenzprodukte durch technische Maßnahmen ebenso wenig erlaubt ist.

Im Kontext dieses Kapitels haben Sega und Lotus versucht, Konkurrenten mit rechtlichem Maßnahmen von ihren Standards auszuschließen, haben aber nicht Recht bekommen. Diese Entscheidungen waren aus wohlfahrtsökonomischer Sicht gerechtfertigt. Warren-Boulton et. al [53] fordern:

...we do not regard the specific policy issues [...] as reflecting a battle between creative producers and later arrivals attempting to free-ride on their work.[...]The debate is between producers who came first and producers seeking to both build on and advance the past work of the others. . .

Diese beiden Fälle haben sich mit dem Copyright-Schutz beschäftigt, der „automatisch“ vergeben wird und zum Schutz von Autoren gedacht ist. Im Gegensatz dazu wird Patentschutz nur auf Antrag und für den Schutz von Erfindungen vergeben [54]. In der Informationstechnologie ist der Unterschied zwischen den beiden Varianten nicht mehr eindeutig sichtbar, Software kann einerseits als Werk von Autoren (Programmierern) und andererseits als patentierbare Erfindung betrachtet werden.

Im Falle des Patentrechts stehen die Entscheidungen schon weniger in Übereinstimmung mit unseren Argumenten für einen geringen Urheberrechtsschutz in Netzwerkindustrien. Prominentestes Beispiel dafür ist Amazon, die ihre Idee des „One-Click Purchasings“ patentieren ließen und per Gerichtsentscheid ihren Konkurrenten Barnes&Nobles dazu zwangen einen zusätzlichen, überflüssigen Schritt in ihre Verkaufstransaktion einzufügen, da die Idee des Kaufs mit *einem* Click patentiert war. Die Genehmigung dieses Patents und der darauf folgende Gerichtsentscheid haben einerseits zu Protesten der Internet-Community und andererseits zu einer Flut an neuen Patentanmeldungen geführt[20].

Eine weitere Möglichkeit für Firmen rechtlichen Schutz für ihre Software oder Teile ihrer Software, wie zum Beispiel Interfaces, zu erreichen liegt in Lizenzvereinbarungen. Auch bei Lizenzen geht der Trend weg vom für Netzwerkindustrien wohlfahrtsoptimalen geringeren Schutz von Software, Standards und Schnittstellen. So ermöglicht zum Beispiel der Uniform Computer Information Transactions Act (UCITA) in den USA sehr weitreichende Lizenzen, die die Nutzung der Software einschränken und zum Beispiel auch Reverse Engineering verbieten können[41], was ja im oben zitierten Fall Sega vs. Accolade vom Gericht gestattet wurde.

Aus diesen vier exemplarischen Fällen lässt sich erkennen, dass die tatsächliche Rechtsprechung nicht immer den netzwerkökonomischen Empfehlungen folgt. Allerdings stehen wir, wie zu Beginn dieses Kapitels zitiert wurde, auch erst am Beginn einer Diskussion, wie das Rechtswesen an die neue „Information Economy“ anzupassen ist.

Was sind nun die Konsequenzen dieser Beobachtungen für unsere Ausgangsfrage, ob Konkurrenten durch rechtliche Maßnahmen von free-ridership ausgeschlossen werden können?

Prinzipiell ist es möglich, wenn auch mit folgenden Einschränkungen:

1. Rechtlicher Schutz kann vor allem dann erwartet werden, wenn für eine Technologie Patentrechte oder gültige Lizenzvereinbarungen bestehen.
2. Falls der rechtliche Schutz zu einer Ausweitung einer marktbeherrschenden Stellung auf Komplementärgüter führt, kann das kartellrechtliche Konsequenzen zur Folge haben. Durch Lizenzierung der Technologie an Konkurrenten zu nicht prohibitiven Gebühren, können solche Konsequenzen vermieden werden.
3. Im Internet und bei anderen digitalen Technologien können Verstöße gegen Schutzrechte unter Umständen nur schwer geahndet werden. Perfekte digitale Kopien, Internationalität und Anonymität erschweren die Umsetzung von Rechtsansprüchen.

Diese drei Punkte lassen sich am Beispiel der CD-Technologie demonstrieren. Da der CD-Standard patentiert ist, können die Patentinhaber Philips und Sony Lizenzgebühren von anderen Firmen verlangen, die CD-Player oder CDs herstellen. Philips und Sony erzeugen CD-Player. Wenn sie die Lizenz für die Erzeugung von CDs nur an die mit ihnen affilierten Musikgesellschaften (Columbia and Polygram) vergeben hätten, hätten sie unter Umständen mit kartellrechtlichen Klagen rechnen müssen.

Schließlich sind solche Lizenzgebühren in der „realen“ Welt auch leichter durchsetzbar, da Schutzverletzungen eher geahndet werden können, wenn physische Distributionswege genutzt werden müssen.

Eine Strategie, die CD-Hersteller aber auch Unternehmen in der Informationstechnologie verwenden ist die Einführung einer starken „Compliance“ Marke, die nur nach einer Kompatibilitätsprüfung und Lizenzierung durch den Markeninhaber geführt werden darf. Unternehmen, die eine Java-Lizenz kaufen, die also nicht nur Java-Klassen entwickeln, sondern in ihre Produkte auch die Java Virtual Machine integrieren wollen, müssen ihr Produkt einem „Compliance-Test“ durch Sun unterziehen, bevor sie die Marke und das Logo „Java-Compatible“ tragen dürfen. Microsoft hat zum Beispiel in seinem Internet Browser zu Beginn eine nicht vollständig kompatible Java Virtual Machines integriert und wurde von Sun unter anderem auf Unterlassung des Führens des „Java-Compatible“ Logos geklagt[30]. Genauso könnte auch eine Marke für Küchengeräte eingeführt werden, die den CookML Standard verstehen. Fremdanbieter die damit werben wollen, dass ihre Geräte CookML-Rezepte lesen können, müssen sich dafür von ModernCooking lizenzieren und ihre Geräte auf vollständige Kompatibilität prüfen lassen. Dadurch sichert sich ModernCooking nicht nur Lizenzannahmen, sondern verhindert auch die Fragmentierung des Standards durch nur teilweise kompatible Implementationen.

Zusammenfassend kann man also feststellen, dass Ausschließbarkeit durch rechtliche Maßnahmen in manchen Fällen möglich sein kann, im Internet teilweise schwer durchsetzbar ist und die rechtliche Lage relativ komplex und im Wandel begriffen ist.

Bis jetzt haben wir nur Kompatibilitätsstandards auf ihre public good Eigenschaften untersucht. Der Zweck dieser Standards ist es Kompatibilität und Interoperabilität zwischen Programmen zu ermöglichen und dazu ist es nötig die Definition des Standards zu veröffentlichen. Dadurch wird die Ausschließbarkeit erschwert, denn wenn eine Standarddefinition allgemein bekannt ist, können Konkurrenten nur schwer davon ausgeschlossen werden, ein Produkt zu

entwickeln, dass diesem Standard gehorcht. In den beiden oben, analysierten Fällen ist die Lage noch deutlicher, da der Standard auf der obersten Ebene jedem offen steht. Jeder soll Rezepte oder Windows-Programme schreiben können und dementsprechend müssen die Standards vollkommen öffentlich zur Verfügung gestellt werden.

Wenn auf die Eigenschaft der Kompatibilität verzichtet wird, ergibt sich ein anderes Bild. Als Beispiel soll ein beliebiges Electronic-Payment System dienen, z.B. SET (Secure Electronic Transactions) oder DigiCash. In unserem Schichtenmodell ergibt sich für alle diese Systeme ein äquivalentes Bild (siehe Abbildung 5.2).

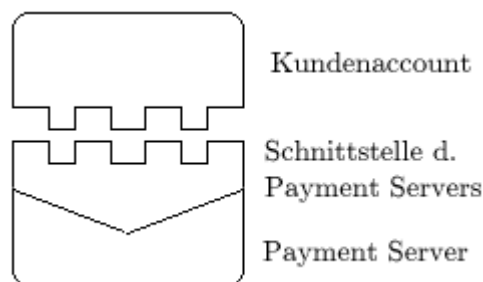


Abbildung 5.2: Schichtenmodell einer klassischen electronic payment Lösung

Die oberste Ebene, die Kundenseite, dient der Identifikation des Kundenaccounts. Je nach Anwendung kann sie z.B. aus einer auf Kundenseite installierten Software (bei SET zum Beispiel als „Wallet“ bezeichnet), einem Zertifikat oder einfach nur aus Benutzernamen und Passwort bestehen.

Der mittlere Layer ist ein Programm auf Serverseite, das mit dem „Wallet“ des Kunden kommunizieren kann bzw. Benutzernamen und Passwort des Kunden versteht. Diese Schicht ist der Standard, der eingeführt werden muss und der Netzwerkexternalitäten erzeugt. Je größer die Verbreitung dieses Standards ist bzw. auf je mehr Webseiten der User seinen Benutzernamen und Passwort eingeben kann um zu bezahlen, umso höher ist der Nutzen für den Anwender der Technologie.

Der entscheidende Unterschied zu Kompatibilitätsstandards ist, dass der mittlere Layer mit dem untersten Layer untrennbar verknüpft ist. Die unterste Ebene sind die durch Eingabe von Benutzernamen und Passwort angestoßenen Abläufe, wie die Authentifizierung und die Zahlungsabwicklung. Die Architektur aller derzeit bestehenden electronic payment Systeme ist darauf ausgelegt, dass die Schnittstelle zur Kundenseite direkt mit den Abläufen der untersten Ebene verbunden ist.

Das ist am Besten durch Betrachtung einer extremen Alternative zu verstehen. Die Abbildung 5.3 zeigt, wie ein e-payment System als Kompatibilitätsstandard implementiert werden könnte.³ Die Kundensicht besteht aus einem Zertifikat, das den Kunden und seine Kontoinformationen identifiziert. Der Kunde legt sein Zertifikat der mittleren Schicht, einer standardisierten Schnittstelle, vor, die auf der untersten Ebene von jeder beliebigen Bank implementiert werden kann. Aufgrund der an der Schnittstelle übergebenen Information weiß die Bank, von welchem Konto sie das Geld abbuchen muss. Das System ist das Äquivalent eines Schecksystems in der realen Welt, bei dem mit Scheck bei jedem Händler bezahlt und der Scheck in jeder Bank eingelöst werden kann.

³Dieses Beispiel dient lediglich der Illustration einer extremen Alternative und deswegen bleiben einige Implementierungsdetails (Sicherheit, Internationalität...) offen.

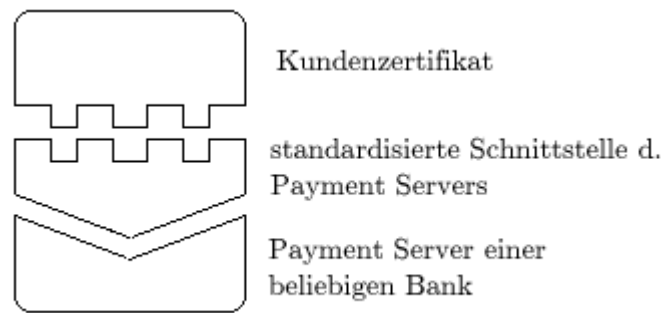


Abbildung 5.3: Schichtenmodell eines Kompatibilitätsstandards für e-payment

Diese zweite e-payment Variante kann im Extremfall so offen gestaltet werden, dass niemand von der Implementation der Schnittstelle ausgeschlossen werden kann. Die Schnittstelle ist ein public good.

Bei der ersten, real umgesetzten Variante hingegen trifft die Eigenschaft der Nicht-Ausschließbarkeit nicht zu. Zum Beispiel kann kein Konkurrent von SET einfach eine Schnittstelle zum SET-Wallet des Kunden aufbauen und die Bezahlung selbst abwickeln, da ihm das Gegenstück zum Kundenwallet auf Serverseite, nämlich der Kundenaccount mit den Kreditkartendaten, fehlt.

Ausschließbarkeit kann also auch dadurch erreicht werden, dass kein offener Kompatibilitätsstandard eingeführt wird, sondern ein System bei dem die einzelnen Schichten untrennbar miteinander verknüpft sind.

Dass dies wirklich in vielen Fällen möglich ist, soll nun noch durch eine Implementation der Kochrezepte illustriert werden, bei der keine free-ridership möglich ist: Die Küchengeräte der Firma ModernCooking können nicht mit jeder beliebigen Webseite, sondern nur mit bestimmten von ModernCooking lizenzierten und durch ein Zertifikat identifizierten Datenbanken kommunizieren. Diese Datenbanken dürfen Rezepte wiederum nur an von ModernCooking zertifizierte Küchengeräte weitergeben. Andere Hersteller können nur Daten aus der Rezeptdatenbank beziehen, wenn sie von ModernCooking eine Lizenz dafür erworben haben. In die Datenbanken werden Rezepte nur über Formulare eingegeben, es gibt keinen Kompatibilitätsstandard, der den Upload in einem bestimmten Format ermöglicht.

Allerdings ist in diesem Fall schon sehr deutlich, dass für das Verhindern von free-ridership ein Preis zu bezahlen ist. Im Kapitel 5.5 wird dieser trade-off genauer beschrieben.

Es wurden also drei Möglichkeiten identifiziert Ausschließbarkeit zu erreichen und free-ridership zu verhindern:

1. hohe Komplexität
2. rechtliche Mittel
3. eng verbundene Layer

Da aber Verhindern der free-ridership in all diesen Fällen immer auch Nachteile mit sich bringt, sollen in den nächsten beiden Kapiteln die Konsequenzen eines Standards mit public good Charakter untersucht werden.

5.3 Free-Ridership und die Teilnahme am Standard-Entwicklungsprozess

Wenn ein Standard ein public good ist, also jedermann kostenlos zur Verfügung steht, wer ist dann bereit in seine Entwicklung zu investieren?

Dieses Kapitel beschäftigt sich mit den Auswirkungen der free-ridership vor dem Zeitpunkt der Einführung des Standards. Wir werden mit Hilfe der Spieltheorie analysieren, ob sich eine Kooperation zur Entwicklung eines public good Standards bilden lässt oder ob alle Beteiligten den Weg der free-ridership vorziehen. Für die Analyse ziehen wir in Anlehnung an Foray[16] das von Schelling 1978 eingeführte „Uniform Multi-Person Prisoner Dilemma“ (UMDP) heran.

Unsere Ausgangslage ist, dass eine Firma einen bestimmten Standard mit public-good Eigenschaften einführt und dafür eine Koalition aus Partnern zur Unterstützung bei der Definition und Einführung des Standards bilden möchte. Den Partnern entstehen Kosten, da sie Ressourcen für die Definitionsarbeiten freistellen und den Standard bei der Einführung durch Marketingaktivitäten unterstützen müssen. Außerdem kann die Kooperation zu einer unerwünschten Bindung führen. Denn kooperiert eine Firma nicht, kann sie bei einem Flop des Standards leichter einen eigenen Weg zur Definition eines neuen Standards einschlagen.

Für unsere Analyse spielt es keine Rolle, ob es sich bei der Kooperation um eine lose Zusammenarbeit oder die Teilnahme an einem formalen Standardisierungsprozess handelt (z.B. einer ISO-Standardisierung).

Jeder potentielle Koalitionsteilnehmer hat zwei Möglichkeiten sich zu entscheiden. Entweder er nimmt an der Kooperation teil (CO) oder er wählt die free-ridership (FR). Der Nutzen für jede der zwei Möglichkeiten ist für alle Firmen gleich. Vorausgesetzt alle Unternehmen dieses Modells sind an einer Einführung des Standards interessiert, so ist der Nutzen für jedes einzelne Unternehmen umso größer, je mehr Unternehmen sich für CO entscheiden. Denn mit der Anzahl der Kooperationspartner steigen auch die Erfolgsaussichten für den Standard. Gleichzeitig ist für jedes einzelne Unternehmen der Nutzen größer, wenn es sich selbst für free-ridership entscheidet, da so der Standard genutzt werden kann, aber keine Kosten für die Einführung getragen werden müssen.

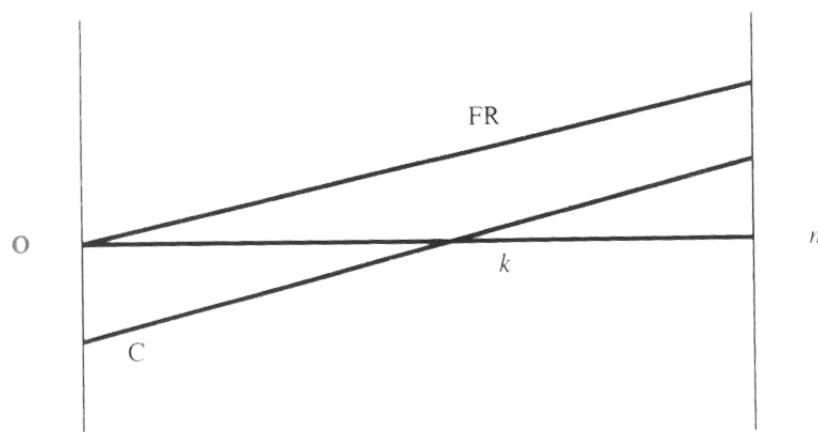


Abbildung 5.4: UMDP-Modell mit einem Gleichgewicht im Nullpunkt, [16]

Dieser Sachverhalt ist in Abbildung 5.4 veranschaulicht. Die beiden mit CO und FR markierten Linien geben den Nutzen für jede einzelne Firma an, wenn sie sich für CO oder FR entscheidet, abhängig von der Anzahl anderer Firmen, die sich für Kooperation entschieden haben. Am linken Ende entscheiden sich 0 Unternehmen für die Teilnahme an der Kooperation, am rechten Ende alle n Unternehmen. Der Abstand zwischen den beiden Linien verringert sich, da die Fixkosten für die Kooperation mit steigender Teilnehmeranzahl auf mehr Unternehmen aufgeteilt werden und die Kosten für jeden Einzelnen damit sinken.

An jedem Punkt der horizontalen Achse ist der Nutzen für ein Unternehmen größer, wenn es sich für FR entscheidet, da die FR-Linie immer über der CO-Linie liegt. Daher entscheiden sich alle Firmen für free-ridership und es kommt zu keiner Kooperation. Oder anders ausgedrückt, das einzige mögliche Ergebnis dieser Situation, das einzige Gleichgewicht, liegt am linken Ende der Grafik im Nullpunkt.

Unter Einbeziehung von Methoden sozialer Organisation gibt es beim UMDP allerdings noch eine weitere Lösung. Wenn mindestens k ($k > 1$) Unternehmen sich entschließen an der Kooperation teilzunehmen, so haben sie einen höheren Nutzen, als wenn auch sie sich für free-ridership entschlossen hätten (siehe Abbildung 5.4). Wenn also durch Kommunikation, Verträge oder andere soziale Mechanismen eine Koalition mit der Mindestgröße k gebildet werden kann, so ist der Nutzen für die Teilnehmer an dieser Koalition größer, als wenn sie nicht teilgenommen hätten. Für die Unternehmen, die sich trotzdem für FR entschieden haben, ist der Nutzen aber noch höher.

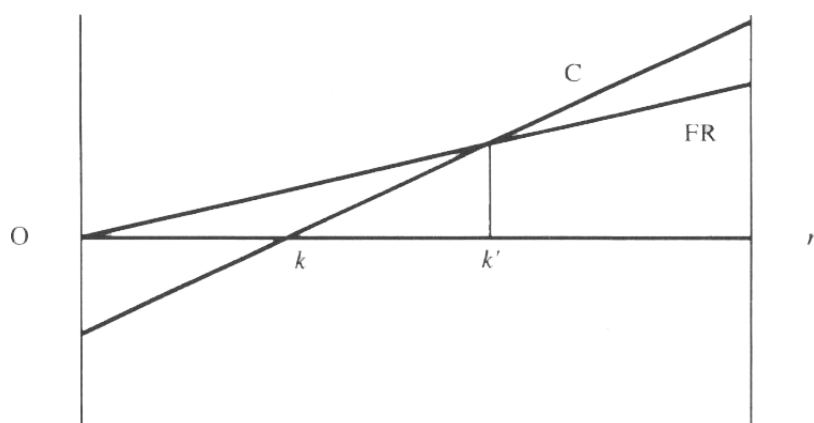


Abbildung 5.5: UMDP-Modell mit Gleichgewichten im Nullpunkt und bei k' , Quelle: [16]

Der erste Ausweg aus dem Dilemma ist es also durch soziale Koordination eine Koalition von ausreichender Größe zusammenzustellen. Der zweite Ausweg lässt sich erkennen, wenn wir die zusätzliche Annahme in das Modell einführen, dass Unternehmen von der Teilnahme an einer Standardentwicklung auch profitieren. Foray[16] zitiert Studien, die gezeigt haben, dass zum Beispiel Lerneffekte und Informationszugewinne solch positive Effekte der Teilnahme an einem Standard-Komitee sind. Weiters profitieren Unternehmen dadurch, dass sie die Richtung des Standards gemäß ihren Wünschen und Geschäftsinteressen beeinflussen können. Zum Beispiel könnte ein Unternehmen, das auf den Bereich gesunde Ernährung spezialisiert ist, bei einer Beteiligung an den Definitionsarbeiten zum CookML-Standard darauf achten, dass der Standard Möglichkeiten zur detaillierten Angabe von Kalorien und Nährwert-Informationen vorsieht. Schließlich bekommen Unternehmen Informationen über Aufbau und Entwicklungsstatus des Standards deutlich früher, wenn sie an der Entwicklungskooperation beteiligt sind. Sie

kennen schon die Testversionen des Standards und haben dadurch einen deutlichen Vorsprung gegenüber free-ridenden Konkurrenten.

Diese positiven Effekte der Kooperation verändern unser Modell entscheidend. (siehe Abbildung 5.5). Nun schneiden sich die Nutzenkurven für die beiden Entscheidungen CO und FR in einem Punkt k' . Dieser Punkt gibt die Mindestgröße einer Koalition an, ab der es für jedes Unternehmen besser ist an der Kooperation teilzunehmen als zu free-riden.

Für zu kleine Koalitionen (links von k') liegt der Nutzen für die FR-Entscheidung noch über CO. Die Koalition ist zu klein um genügend positive Informations- und Lerneffekte zu generieren, um die Kosten von CO auszugleichen. Außerdem ist bei dieser kleinen Koalitionsgröße das Risiko höher, dass der Standard nicht durchgesetzt werden kann. Rechts von k' überwiegen hingegen die positiven Effekte die Kosten der Kooperation.

Eine weitere Variante ist in Abbildung 5.6 dargestellt. Hier schneiden sich die beiden Nutzenkurven ein zweites Mal im Punkt k'' . Das berücksichtigt den Umstand, dass eine Koalition auch zu groß werden kann. Bei zu vielen Teilnehmern sinkt der Einfluss des Einzelnen bei Implementationsentscheidungen, steigen die Koordinationskosten und steigt die Gefahr, dass Unternehmen auf Grund der positiven Effekte an der Koalition teilnehmen, aber dann wenig zum Entwicklungs- und Einführungsprozess beitragen. Es entstehen also free-rider innerhalb der Koalition, die bei großen Gruppen weniger leicht aufspürbar sind.

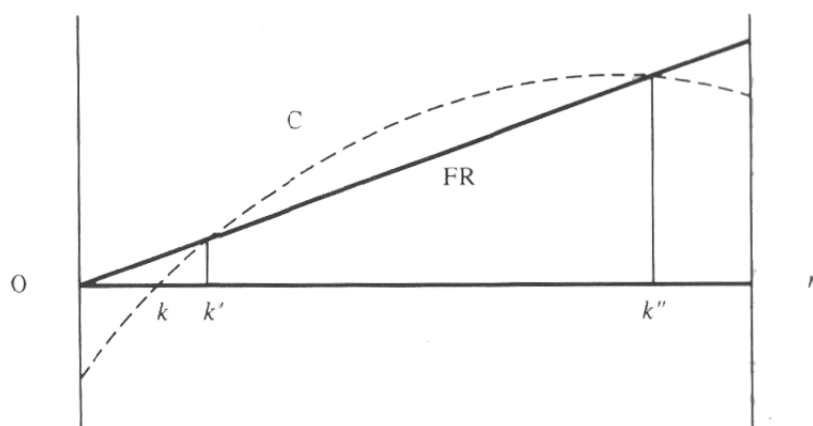


Abbildung 5.6: UMDP-Modell mit Gleichgewichten bei 0 und k'' , Quelle: [16]

Zu welchen Ergebnissen, oder ökonomisch gesagt Gleichgewichten, führen nun die beiden erweiterten Modelle? In dem Modell der Abbildung 5.5 liegen die beiden Gleichgewichte an den linken und rechten Endpunkten. Die Grenze zwischen den beiden Gleichgewichten, der „tipping point“, liegt bei k' . Ob eine Koalition aller Teilnehmer oder gar keine Koalition entsteht, ist abhängig von den Ereignissen während der Phase der Koalitionsbildung. Ist die erwartete Anzahl von Mitgliedern der Koalition größer als k' , so werden schließlich auch alle anderen Unternehmen kooperieren, da rechts des tipping points der Nutzen für Kooperation höher ist. Bleibt die erwartete Anzahl von Mitgliedern hingegen unter k' , so wird sich die Koalition letztendlich wieder auflösen, da der Nutzen für jeden Einzelnen größer ist, wenn er sich für die free-ridership entscheidet.

Ist es möglich durch soziale Mechanismen eine kleinere Koalition von Mindestgröße k zu erzeugen (siehe oben), so liegt das linke Gleichgewicht nicht bei 0 sondern zwischen k und k' .

In der Variante des Modells mit zwei Schnittpunkten (Abbildung 5.6) gibt es ebenfalls zwei Gleichgewichte, das erste liegt analog zur Abbildung 5.5 wieder beim Nullpunkt bzw. auf der Strecke zwischen k und k' . Das zweite Gleichgewicht liegt bei k'' . Denn jedes Unternehmen, das überlegt zu einer Koalition von k'' Teilnehmern dazuzustoßen, wird sich für FR entscheiden, da dafür der Nutzen rechts von k'' größer ist. Jedes Unternehmen, das überlegt einer Koalition mit einer Größe zwischen k' und k'' beizutreten wird sich für den Beitritt entscheiden, da die CO-Nutzenkurve über der FR-Kurve liegt. Der tipping point liegt wieder bei k' .

Zusammenfassend können wir aus dieser Analyse die folgenden Schlüsse ziehen:

1. Die Trägheit kann durch soziale Mechanismen überwunden werden, wenn es gelingt eine Koalition von der Mindestgröße k zu bilden.
2. Wenn die Beteiligung an der Koalition positive Effekte verspricht, führt ein public good Standard nicht automatisch zur Nicht-Kooperation. Ein Unternehmen, das eine Koalition bilden will, sollte darauf achten, dass solche positiven Effekte entstehen und dass sie den potentiellen Koalitionsteilnehmern bewusst sind. Zum Beispiel könnte man das Know-How, das einzelne Teilnehmer in die Koalition einbringen, verdeutlichen, um Lerneffekte aufzuzeigen. Außerdem kann der Informationsvorsprung, den Teilnehmer der Koalition gegenüber der free-ridingen Konkurrenz haben, vergrößert werden, zum Beispiel indem den Teilnehmern vor der Veröffentlichung des Standards ausreichend Zeit zur Produktentwicklung gegeben wird.
3. Der Erfolg der Koalitionsbildung ist davon abhängig ob bestimmte Grenzen (k und k') überschritten werden können. Damit spielen soziale Interaktion, Überzeugungskraft und Erwartungshaltung eine wesentliche Rolle. Die Überzeugung starker Partner zu Beginn erscheint besonders wichtig, da sie das Vertrauen in die Koalition stärken, eine hohe Durchsetzungswahrscheinlichkeit des Standards versprechen und durch Know-How positive Informationseffekte schaffen. Auch ein schon bestehendes Netzwerk von engen Kooperationspartnern kann der innovierenden Firma helfen, die anfängliche Trägheitsschwelle zu überwinden.

Nachdem die Hürde der erfolgreichen Einführung genommen wurde, gilt es mit dem Standard direkt oder indirekt Geld zu verdienen. Wie das bei kostenlos verfügbaren, offenen Standards möglich sein kann behandelt das nächste Kapitel.

5.4 Finanzielle Verwertung von public-good Standards

Die Betrachtung in diesem Kapitel beschränkt sich auf Standards, die vollständige public goods sind. Es werden weder Konkurrenten von dem Standard ausgeschlossen noch Lizenzgebühren für die Verwendung des Standards eingefordert.

Prinzipiell ist es möglich aus öffentlichen Standards Gewinn zu lukrieren. In dem oben beschriebenen Beispiel (S. 33) eines offenen electronic payment Systems liegt ein public-good Standard vor und trotzdem könnten die beteiligten Banken Gebühren für die Transaktionen einnehmen und so von dem Standard profitieren. Allerdings ist die Konkurrenz deutlich größer als bei den „herkömmlichen“ electronic payment Varianten, da durch die Nicht-Ausschließbarkeit Jeder freien Zugang zum Markt hat.

Public-good Standards verhindern also nicht die finanzielle Verwertbarkeit, erhöhen aber die Konkurrenz.

Das erste Argument aus finanziellen Gründen einen public-good Standard einzuführen bringen Shapiro und Varian mit der folgenden prägnanten Formel auf den Punkt[48]:

$$Your\ reward = Total\ value\ added\ to\ industry \times your\ share\ of\ industry\ value$$

Der entscheidende Punkt ist, dass Unternehmen selbst davon profitieren, wenn sie durch einen Standard ihr Industriesegment beleben. Intel hat zum Beispiel den „accelerated graphics port (AGP)“ als offenen Kompatibilitätsstandard eingeführt, um den Markt für Computergrafik anzukurbeln und damit die Nachfrage nach ihren MMX-Pentium Prozessoren zu erhöhen. Die Kontrolle über die Multimediaspezifikation MMX der Prozessoren haben sie hingegen behalten.

Genauso wird die Einführung eines Rezeptstandards in erster Linie die Haushaltsgeräteindustrie ankurbeln. Die Möglichkeit Rezepte aus dem Internet herunterzuladen und von Kühlschränken oder Küchengeräten interpretieren zu lassen erhöht Nachfrage und Preis jener neuen Geräte, die diese Technologie beherrschen.

Kann durch einen Standard nur der erste Teil der Formel, der Wert der ganzen Industrie erhöht werden, so profitieren die free-ridenden Konkurrenten genauso wie das Unternehmen, das den Standard eingeführt und dafür die Kosten getragen hat. Das einführende Unternehmen wird auf jeden Fall seinen Anteil am Wachstum des Marktes einfahren, zusätzlich aber auch versuchen durch seine Rolle als Standardentwickler den zweiten Faktor der Formel zu beeinflussen: seinen eigenen Marktanteil.

Adobe ist es mit seinem Portable Document Format (PDF) ausgezeichnet gelungen, einen vollkommen offenen Standard mit public-good Charakter einzuführen, selbst den größten Marktanteil zu halten und so mit dem offenen Standard Gewinn zu erwirtschaften. Abbildung 5.4 zeigt den Aufbau im Schichtenmodell. Die mittlere Ebene ist der PDF-Standard, der öffentlich zugänglich ist und von jedermann ohne Entrichtung von Lizenzgebühren benutzt werden kann. Die untere Ebene wird von den Anwendungen gebildet, mit denen PDF-Dokumente erstellt werden können, die oberste von Anwendungen zum Anzeigen der PDF-Dokumente. Beide Ebenen sind potentiellen free-ridern frei zugänglich.

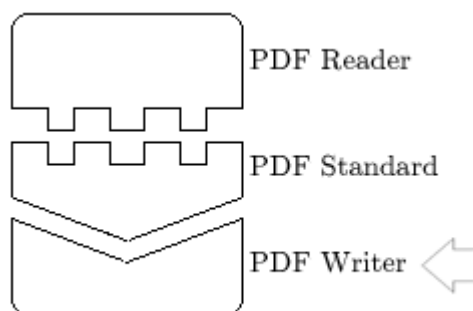


Abbildung 5.7: Schichtenmodell des Adobe PDF Standards, der Pfeil zeigt an, auf welcher Schicht Adobe Geld verdient

Auf der obersten Ebene stellt Adobe ein kostenloses Produkt zur Anzeige von PDF-Dokumenten zur Verfügung, den Acrobat Reader. Dadurch konnte das PDF-Format seine große Verbreitung erreichen. Da der Acrobat Reader ein hochwertiges, kostenloses Produkt ist, hat kein anderes Unternehmen Interesse daran auf dieser Ebene freezuriden. In Konkurrenz zum Acrobat Reader ist kein Profit zu erzielen. Auf der untersten Ebene bietet Adobe ein kostenpflichtiges, (rund 370 EUR teures) Produkt zur Erstellung von PDF-Dokumenten an. Hier schlägt Adobe Profit aus dem PDF-Standard und hier wäre es auch für andere Unternehmen interessant ein

Konkurrenzprodukt anzubieten. Durch den kostenlosen und weit verbreiteten Acrobat Reader ist es Adobe jedoch gelungen seinen Firmennamen untrennbar mit dem PDF-Format zu assoziieren und sich damit auch den Großteil des Marktes auf der untersten Ebene zu sichern. In Marktsegmenten, an denen Adobe kein Interesse hat, zum Beispiel weil dort Produkte traditionell kostenlos angeboten werden, bieten andere Anbieter Anwendungen zum Erstellen von PDF-Dokumenten an und erhöhen dadurch den Wert des PDF-Netzwerks (z.B. die verschiedenen PDF-Konverter für LaTeX).

Adobe hat also den Komplementäreffekt zwischen zwei Produkten ausgenutzt, die Verbreitung des einen steigert die Nachfrage nach dem anderen. Seinen Marktanteil auf der unteren Ebene hat Adobe durch das auf der obersten Ebene gewonnene Image erhöht.

Auch Suns Java Strategie baut darauf, das durch Java gewonnene Image auf Komplementärbereiche auszudehnen. Die Schichtenstruktur der Java-Technologie sieht jedoch anders als die der bisher analysierten Standards aus (Abbildung 5.4). Auf der untersten Ebene liegt die Java Virtual Machine, die den Java Code interpretiert und ausführt. Diese Virtual Machine interpretiert nur Code, der der darüberliegenden Spezifikation der Java-Sprache entspricht. Zusätzlich existiert hier noch eine weitere Schicht, die von Sun zur Verfügung gestellten Standard-Java Klassen. Diese Klassen müssen auf jedem Java-System verfügbar sein, müssen aber nicht genutzt werden. Genauso könnte ein Alternativenanbieter alternative Systemklassen entwickeln und zur Verfügung stellen.

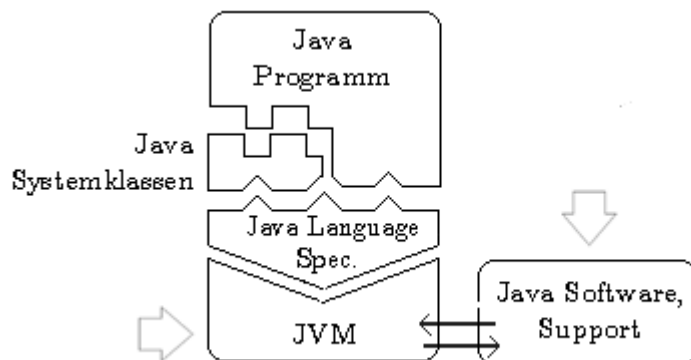


Abbildung 5.8: Schichtenmodell des Java Standards, die dicken Pfeile zeigen an, auf welchen Schichten Sun Geld verdient

Wie schon oben erwähnt, ist die unterste Ebene der JVM für Anwender und Programmierer frei verfügbar, nur wer sie in eines seiner Produkte integrieren will (z.B. in einen Webbrowser), muss eine Lizenz erwerben. Das ist die erste Art für Sun durch den Standard Geld zu verdienen, deren Bedeutung aber von Sun heruntergespielt wird [46]. Im Gegensatz zu den bisher analysierten Standards, CookML, PDF und Windows API, ist es nicht Suns Ziel auf dieser untersten Ebene durch Komplementäreffekte Gewinne zu erwirtschaften. Der Großteil der User verwendet auch die unterste Ebene kostenlos. Die wesentlich wichtigere Einnahmequelle stellen Java-basierte Produkte und Dienstleistungen dar. Sun hat sich durch die Einführung von Java den Ruf gesichert *das* Java-Unternehmen zu sein, sie haben bewiesen, dass sie die innovative Stärke haben eine revolutionäre Technologie einzuführen und sie haben die Loyalität tausender Softwareentwickler gewonnen. Dennoch ist es ihnen nach einer Studie der Gartner Group

nicht gelungen dieses Image in hohe Marktanteile bei Java-Komplementärprodukten umzusetzen. Nach Suns eigenen Angaben ist die Java-Technologie ein Schlüsselfaktor für 90 % von Suns Verkäufen. Dennoch haben andere Unternehmen von Java mehr profitiert als Sun selbst. IBM und BEA sind Marktführer bei Java-basierten Applikationsservern, Sun hält mit 7% Marktanteil nur den dritten Platz. Suns Schwäche wird mit strategischen Fehlentscheidungen zu Beginn der Java-Einführung begründet und auch darauf zurückgeführt, dass die anderen Unternehmen schon vor der Einführung von Java in diesem Bereich tätig waren, während Sun erst mit Java massiv in den Software-Markt eingestiegen ist[46].

Sun hat von Java zweifellos profitiert, aber das gewonnene Image nicht so gut auf verwandte Produkte umsetzen können wie Adobe. Die Bindung zwischen dem Java-Standard und den von Sun angebotenen Produkten und Serviceleistungen ist auch nicht so hoch wie bei Adobe. Wie in den Abbildungen 5.4 und 5.4 ersichtlich, lukriert Adobe Gewinn im direkten Komplementärprodukt des PDF-Standards, während Suns Gewinn, abgesehen von den Lizenzgebühren, in relativ lose an den Standard gekoppelten Produkten und Dienstleistungen entsteht.

Neben Imagebildung und Vermarktung von Komplementärgütern gibt es noch einen dritten Mechanismus durch die Einführung eines offenen Standards den Marktanteil zu erhöhen. Oft verschafft sich die Standard-entwickelnde Firma einen Vorsprung, indem sie ihren Wissensvorsprung über den Standard ausnützt, bessere Produkte entwickelt und diese schneller auf den Markt bringt als die Konkurrenz. Das reicht von kleinen Tricks, wie sie Adobe für sich behalten hat, um bei der Erstellung von PostScript Dokumenten eine bessere Darstellung zu erzielen [48], bis zu wettbewerbsrechtlich anfechtbaren Methoden, wie sie Microsoft für ihr Windows-API verwendet. Kritiker behaupten, dass Microsoft Teile seines Windows-API geheim hält bzw. zuerst an interne Abteilungen weitergibt, bevor sie veröffentlicht werden, obwohl Microsoft das Windows-API zu einem offenen Standard erklärt hat. Microsoft nützt also das Wissen aus der Entwicklung des API-Standards, um besser und schneller Windows Anwendungen programmieren zu können als die Konkurrenz.

Diese Fallstudien haben häufige Strategien zur finanziellen Verwertung von public-good Standards aufgezeigt, aber die Probleme Suns im Applikationsserver-Markt haben auch verdeutlicht, dass gerade hier eine detaillierte Strategie nur für jeden einzelnen Fall entwickelt werden kann.

5.5 Konkurrenten ausschließen oder free-rider akzeptieren?

Diese Fragestellung ist eng mit anderen später behandelten Themen verknüpft. Einen public-good Standard einzuführen bringt zum Beispiel ähnliche Vorteile, wie sie im Kapitel 7 für offene Standards behandelt werden. Ob es vorteilhaft ist Konkurrenten den Zugang zum Standard zu ermöglichen, ist wiederum eng mit der Frage der Kooperation (s. Kapitel 8) verbunden.

Deshalb wird der trade-off zwischen dem Ausschließen von Konkurrenten vom Standard und den dadurch entstehenden Nachteilen hier relativ kurz und vor allem in zwei Fragen behandelt: Welche Nachteile können aus dem Verhindern der free-ridership entstehen? Welche Faktoren müssen vorliegen, um diese Nachteile auszugleichen und das Verhindern der free-ridership trotzdem effizient erscheinen zu lassen?

Der trade-off lässt sich wieder am Besten mit der schon oben zitierten Formel

$$\textit{Your reward} = \textit{Total value added to industry} \times \textit{your share of industry value}$$

beschreiben. Das Ziel muss es sein den eigenen Gewinn zu maximieren, anstatt nur den Zugang zum Standard möglichst restriktiv zu gestalten um Konkurrenten von der free-ridership abhalten zu können. Denn sonst besteht die Gefahr ein sehr großes Stück von einem sehr kleinen Kuchen

zu erobern, obwohl ein kleines Stück von einem größeren Kuchen mehr Gewinn eingebracht hätte.

Das könnte zum Beispiel einer der Gründe für das Scheitern zahlreicher electronic payment Systeme gewesen sein (z.B. Millicent, eCash), denen es trotz technischer Überlegenheit gegenüber der derzeit am weitesten verbreiteten Kreditkartenzahlung nicht gelungen ist, sich am Markt zu behaupten. Alle entwickelten Systeme versuchten die mittlere und untere Ebene (s. Abbildung 5.2) für sich zu monopolisieren, anstatt ein offenes System, wie in Abbildung 5.3 skizziert, oder zumindest ein halboffenes System, das Zugang zu diesen Ebenen über Lizenzierung ermöglicht, zu implementieren.

Durch den Ausschluss von Konkurrenten können folgende Nachteile für den Standard entstehen:

- Der erste und gravierendste Nachteil ist der Verlust von Netzwerkexternalitäten für die User, da sich das Netzwerk verkleinert. Wenn zum Beispiel nur eine Firma CookML-Geräte anbietet verkleinert sich das Netzwerk in dem User Kochrezepte austauschen können.
- Auch die für die Verbreitung eines Standards wichtigen Konsumentenerwartungen werden durch Ausschließen anderer Anbieter beeinträchtigt. Gerade in der kritischen, offene Standards gewöhnten Internet-Community besteht Misstrauen gegenüber Unternehmen, die versuchen eine Technologie für sich zu monopolisieren.
- Die oben beschriebenen Beispiele, wie durch Verknüpfung zweier Layer free-rider verhindert werden können (siehe Seite 33, illustrieren auch, dass Ausschließen von Konkurrenten eine Adaption des Aufbaus des Standards erfordern kann, die die Funktionalität des Gesamtprodukts entscheidend einschränkt.
- Am Beispiel von Adobes PDF haben wir gesehen, dass free-riden in Bereichen in denen ein Unternehmen keinen Profit machen kann (z.B. in der Umgebung von open source software) oder keine Kompetenz hat (z.B. auf einem seltenen Betriebssystem) durchaus erwünscht ist, da es die Verbreitung des Standards erhöht, ohne profitable Marktbereiche zu beschneiden. Ein offenes e-payment System ermöglicht zum Beispiel auch die Erschließung von Ländern in denen die einführende Firma wenig Marktmacht hat, da die lokalen Banken die Einführung übernehmen können.

All diese Faktoren mindern den Nutzen des Standards für den Konsumenten, was zu einer geringeren Verbreitung führt und die Durchsetzung des Standards auch gänzlich verhindern kann. Sie senken den ersten Teil der zitierten Formel, den Mehrwert, der durch den Standard der Industrie hinzugefügt wird.

Wenn Unternehmen diese Nachteile dem positiven Effekt des gesteigerten Marktanteils gegenüberstellen, sollten sie untersuchen, ob die folgenden Faktoren zutreffen, die die Bedeutung der Nachteile lindern können:

- Eine vorhandene, große installed base, die zur Verbreitung des neuen Standards genutzt werden kann, gleicht den Nachteil der gesunkenen Netzwerkexternalitäten aus. Ein mit Microsoft Office lesbares Dateiformat kann zum Beispiel eher restriktiv gehalten werden als eines mit einer kleineren installed-base an Viewern. Durch die große installed base der Office-Suite sinkt der verlorene Nutzen, das Format nicht auch auf Konkurrenzprodukten anzeigen zu können.

- Das Vorhandensein „natürlicher“ Ausschließungsmechanismen verhindert, dass die Produktfunktionalität zu sehr eingeschränkt werden muss. Das trifft zum Beispiel für komplexe, schwer imitierbare Produkte (unser Beispiel dazu war das Windows Betriebssystem) oder patentierbare Technologien (z.B. der CD-Standard), die durch rechtliche Maßnahmen geschützt werden können, zu.
- Geringe „switching costs“ erhöhen die Bereitschaft von Konsumenten den Einstieg in die Technologie zu riskieren. Ein Dokumentstandard im Internet kann durch den Download eines Plugins relativ leicht und risikolos angenommen werden. Allerdings wird es sich ein User zweimal überlegen, ob er sich einen Küchencomputer kauft, wenn kompatible Geräte nur von einer Firma angeboten werden dürfen.
- Wenn Kompetenz in allen wichtigen Bereichen (z.B. auf allen Betriebssystemen) vorhanden ist, gehen bei der Ausschließung von Konkurrenten keine Marktanteile verloren.
- Ein eingeführter Markenname oder die Unterstützung einer großen Firma steigern das Vertrauen in eine Lösung.

Die Entscheidung, ob Konkurrenten ausgeschlossen werden sollen, ist selten eine binäre Ja/Nein Entscheidung. Verschiedene Modelle einen Standard zu lizensieren oder Mitbewerber in bestimmten Bereichen oder auf bestimmten Schichten des Standards zuzulassen ergeben ein Spektrum an Möglichkeiten, aus denen anhand der in diesem Kapitel beschriebenen Faktoren, diejenige ausgewählt werden soll, die das Produkt *Mehrwert der Industrie* \times *Marktanteil* maximiert.

Kapitel 6

Switching Costs und Lock-In

„Switching costs“ sind jene Kosten, die auftreten, wenn ein Konsument von einem Produkt oder Standard zu einem anderen wechselt. Das Interessante an diesen Kosten ist, dass eine Entscheidung für ein Produkt in der Gegenwart zukünftige Investitionsentscheidungen beeinflusst. Denn bei der nächsten Wahl eines ähnlichen oder eines kompatiblen Produktes entstehen im Falle eines Wechsels zu einer anderen Produktfamilie switching costs. Entscheidet man sich etwa beim Kauf des ersten Computers für einen PC der Windows-Intel Familie, so treten bei der nächsten Wahl eines Computers erhebliche switching costs auf, falls man zum Apple-Standard wechseln will. Schon erworbene Software, Training und angelegte Daten gehen verloren.

Switching costs treten in unterschiedlichen Höhen auf und können den Nutzenvorteil des Wechsels zu einer neuen Technologie übersteigen. Wenn die switching costs so stark sind, dass sie Anwender davon abhalten die Technologie zu wechseln, spricht man von einem „Lock-In“ in die Technologie.

Anwender müssen daher bei der Wahl einer Technologie die zukünftigen switching costs und die Gefahr des Lock-Ins in ihre Überlegungen miteinbeziehen. Unternehmen werden versuchen switching costs für ihre Produkte möglichst hoch anzusetzen um Kunden an ihre Technologie zu binden.

Bevor wir uns einer detaillierten Analyse zuwenden, soll der Begriff der switching costs noch etwas eingegrenzt werden. Switching costs treten in irgendeiner Form für fast jedes Produkte auf. Bereits beim Wechsel des Supermarktes ist ein Konsument mit switching costs konfrontiert, da er sich neu zurechtfinden muss, Stammkunden-Treuepunkte verliert usw. Dieses Kapitel soll sich aber weniger mit den switching costs beschäftigen, die auch in der traditionellen Ökonomie auftreten und schon lange Bestandteil von Marketingstrategien sind, sondern vor allem mit jenen switching costs, die erst durch den Einsatz von Standards entstehen. In Kapitel 6.1 werden dieser Unterschied und die Entstehung von switching costs genauer herausgearbeitet, bevor im Kapitel 6.2 Strategien für die Einführung von Standards mit switching costs diskutiert werden.

Die Begriffe switching costs und Lock-In werden in einer anderen Bedeutung auch oft in Zusammenhang mit Netzwerkexternalitäten verwendet. Wenn Netzwerkexternalitäten vorliegen, kann es, wie oben besprochen, zum Beispiel durch einen first-mover advantage zu einer frühzeitigen Festlegung auf einen Standard kommen. Da der Konkurrenzstandard ein zu kleines Netzwerk hat, können die Anwender nicht wechseln, sie sind im mächtigeren Standard „eingesperrt“ (=locked in). Allerdings handelt es sich hier um kollektiven Lock-In und um kollektive switching costs. Diese kollektiven switching costs sind die Kosten der Koordination eines gleichzeitigen Wechsels aller User zu einem anderen Standard und nicht die Kosten des Wechsels für einen Einzelnen. Die Form des (individuellen) Lock-Ins, mit der sich dieses Kapitel beschäftigt, kann auch auftreten, wenn zwei gleich starke Standards neben einander existieren. In diesem Fall können für einen User, unabhängig von etwaigen Netzwerkexternalitäten, so hohe individuelle switching costs bestehen, dass er nicht wechseln kann. Der durch Netzwerkexternalitäten verursachte kollektive Lock-In tritt im Falle zweier gleich großer Netzwerke allerdings nicht auf. Weiters hat ein neu am Markt eintreffender Konsument keine individuellen switching costs, da er noch in keine Technologie investiert hat. Allerdings ist er ebenfalls vom kollektiven Lock-In in eine Technologie betroffen, weil nur die Technologie mit dem größeren Netzwerk für ihn in Frage kommt.

Der tatsächliche Zusammenhang zwischen (individuellen) switching costs und Netzwerkexternalitäten besteht darin, dass der Nutzen einer neuen Technologie gesenkt wird, wenn Anwender switching costs überwinden müssen, um sie zu adaptieren. Wenn dieser Nutzen wegen des anfangs kleinen Netzwerkes ohnehin schon gering ist, erhöhen switching costs die Gefahr von excess inertia weiter.

6.1 Entstehung von Switching Costs

Shapiro und Varian[48] definieren sieben mögliche Ursachen für switching costs. Von diesen Ursachen sollen zuerst die drei für uns unmittelbar relevanten, durch Standards verursachten, herausgegriffen werden:

- Information und Datenbanken: Wenn Anwender eine große Menge an Informationen im Format eines Standards gespeichert haben, so fallen switching costs an, wenn sie zu einem Produkt wechseln wollen, das diesen Standard nicht unterstützt. Entweder die bestehenden Daten werden konvertiert, wobei Kosten und Qualitätsverluste auftreten können, oder gehen gänzlich verloren. Anwender, die zum Beispiel eine große Menge an VHS-Videos haben, verlieren diese Daten, wenn ihr VHS-Videorecorder defekt wird und sie deshalb zum DVD-Standard wechseln. Switching costs durch Datenbanken steigen mit der Zeit, da die Menge der gespeicherten Informationen mit der Zeit zunimmt.
- Langlebige Güter: Beim Kauf von Produkten mit langer Lebensdauer entstehen für Konsumenten erhebliche switching costs, wenn sie zu einem anderen Produkt wechseln wollen, da die Investition in das erste Produkt damit verloren geht. Doch erst durch die Komplementarität zwischen zwei Gütern wird dieser Lock-In für Anwender gefährlich und für Hersteller interessant. Nämlich dann, wenn Anwender zu dem langlebigen Produkt kompatible Güter kaufen (müssen). So ist es zum Beispiel Iomega's Strategie Laufwerke für ihren proprietären Zip-Standard sehr billig zu verkaufen und ihre Einnahmen aus den Verkäufen der dazu kompatiblen Speichermedien zu beziehen[48]. Die Laufwerke sind dauerhafte Güter und Anwender sind im Zip-Standard so lange gefangen, bis das Laufwerk abgeschrieben, veraltet oder defekt ist.

Da langlebige Güter mit der Zeit ihren Wert verlieren, sinken auch die daraus entstehenden switching costs. Wenn das System allerdings aus mehreren langlebigen Komponenten besteht, bietet sich für Konsumenten kein geeigneter Zeitpunkt zum Wechsel des Systems, da nicht alle langlebigen Güter gleichzeitig ersetzt werden müssen. Auch im Falle der Zip-Laufwerke nehmen die switching costs nicht unbedingt ab, da Informationsbestände auf Zip-Medien aufgebaut werden, die auch wenn das Laufwerk ausgetauscht wird noch switching costs erzeugen. Sie schaffen für Konsumenten den Anreiz, ein zu dem alten Standard kompatibles neues Laufwerk zu kaufen. Es besteht eine Komplementarität zwischen einem langlebigen Gut und Informationsbeständen.

- Training: Für viele Technologien wird eine Ausbildung oder zumindest eine Einlernzeit benötigt. Switching costs fallen an, wenn ein Anwender zu einer anderen Technologie wechselt, die er wieder erst erlernen muss. Da die Erfahrung und das Know-How im Umgang mit einem Standard mit der Zeit zunehmen, steigen meist auch die switching costs mit der Zeit an. Allerdings können switching costs ab einem gewissen Ausbildungsniveau auch wieder sinken, da fundiertes Verständnis einer Technologie das Erlernen einer neuen Technologie mit ähnlichen Grundkonzepten erleichtern kann.

Im Kapitel 4.3 wurde bereits das Beispiel von Microsoft genannt, das versuchte die switching costs von Word Perfect zu Microsoft Word zu minimieren. Viele Fluglinien legen Wert darauf, alle Flugzeuge ihrer Flotte von einem Hersteller zu beziehen und für ihre gesamte Flotte ein standardisiertes Cockpit-Layout zu haben, um den Schulungsaufwand für ihr Personal zu reduzieren[48].

Kurz sollen auch noch die vier weiteren standard-unabhängigen Ursachen für switching costs beschrieben werden:

- Vertragliche Verpflichtungen: Zum Beispiel durch Verträge, die ein Minimum an Einkäufen pro Jahr vorschreiben oder die Unternehmen zwingen die gesamte Ausrüstung von einem Hersteller zu beziehen.
- Spezialanfertigungen: Reparaturen, Erweiterungen, Verbrauchsmaterialien für speziell angefertigte Produkte können oft nur von der Herstellerfirma bezogen werden.
- Kosten der Produktsuche: Ein neues Produkt oder einen anderen Anbieter zu suchen erzeugt sowohl für Unternehmen als auch für deren Kunden switching costs. Beim Wechsel einer Bank müssen Unternehmen Informationen über die Kreditwürdigkeit eines Kunden einholen und Kunden unter komplizierten unterschiedlichen Angeboten den billigsten Anbieter auswählen.
- Treueprogramme: Treueprogramme, wie sie von Supermärkten, Handy-Anbietern und Fluglinien angeboten werden, senken die Preise für Kunden, je mehr Produkte sie bei einem Unternehmen beziehen. Wechselt ein Kunde den Anbieter, so beginnt er im Treueprogramm wieder von null, seine switching costs sind also die verlorenen Treuepunkte. Durch Treueprogramme werden switching costs künstlich erzeugt.

Je nach Offenheit des Standards kann man unterschiedliche Arten des Lock-In unterscheiden. Bei offenen Standards liegt nur ein Lock-In auf einen speziellen Standard vor, Anwender können aber damit rechnen, zwischen unterschiedlichen Anbietern des Standards wählen zu können. Die große Gefahr liegt im Lock-In auf einen proprietären Standard und in der Abhängigkeit von einem einzelnen Unternehmen, das bis zu einem gewissen Ausmaß als Monopolist agieren kann.

Switching costs und Lock-In können, wie auch schon die free-ridership, auf unterschiedlichen Ebenen des Standards entstehen. Beim Rezeptstandard entstehen switching costs auf zwei Ebenen (Abbildung 6.1). Auf der obersten Ebene bauen Anwender Informationsbestände in Form von Rezeptdatenbanken auf. Auf der untersten Ebene werden langlebige Küchengeräte angeschafft. Durch diese beiden Faktoren erfahren Anwender einen Lock-In im Rezeptstandard auf der mittleren Ebene. Hier sieht man, wie die switching costs von Informationsbeständen und langlebigen Gütern sich gegenseitig verstärken. Wenn ein neues Küchengerät angeschafft wird, können Anwender für die Lebensdauer des Gerätes nicht zu einem anderen Standard wechseln. In dieser Zeit erstellen die Anwender Rezepte im Format des Standards. Wenn die switching costs des Küchengerätes gegen null gehen, weil das Gerät ausgetauscht werden muss, haben die Anwender bereits eine Rezeptsammlung aufgebaut, die dann ihrerseits den Lock-In im Standard aufrecht erhält.

Diese Kombination von sich gegenseitig verstärkenden switching costs auf zwei Ebenen tritt häufig auf. Ein Beispiel, bei dem switching costs nur durch Informationsbestände, also auf der obersten Ebene, erzeugt werden, sind E-Mail Programme. Die meisten E-Mail Programme sind entweder gratis oder werden mit dem Betriebssystem mitgeliefert, es entstehen also keine switching costs durch Investitionen in langlebige Produkte. Nur die im Format des Programms abgespeicherten E-Mails und Adressbücher verursachen einen Lock-In für den Anwender. Viele Programme senken die switching costs zum eigenen Standard, indem sie eine Importfunktion für Mailboxes anderer E-Mail Software anbieten. Interessant ist, dass zum Beispiel der Netscape Communicator nur eine Import- aber keine Exportfunktion für E-Mails anbietet, um nicht auch die switching costs beim Verlassen des Standards zu senken.

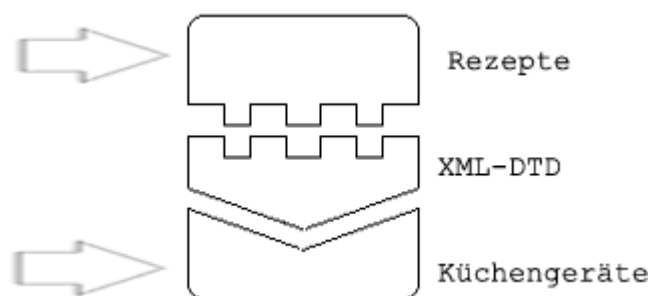


Abbildung 6.1: Switching costs treten auf der obersten und der untersten Ebene auf

Doch Standards, in der dieser Arbeit zu Grunde liegenden Bedeutung, führen nicht zwingendermaßen zu Lock-In. Im Falle eines e-payment Standards treten zum Beispiel nur sehr geringe individuelle switching costs auf. Es wird keine größere Menge an Information gespeichert, zur Verwendung wird kaum Training benötigt und die einzige Investition ist der Zeitaufwand, der für die Einrichtung des Systems benötigt wird.¹

6.2 Folgen der Switching Costs für Standard-Strategien

Nun sollen Maßnahmen zur Einführung von Standards unter Berücksichtigung von switching costs näher beleuchtet werden.

Wie schon bei Netzwerkexternalitäten kann es auch bei switching costs notwendig sein, Investitionen zu tätigen um eine installed base aufzubauen. Für solche Investitionen gibt es zwei Motive. Erstens müssen Anwender, die von einem alten zum neu eingeführten Standard wechseln, switching costs überwinden. Durch sponsorship, wie Rabatte oder Gutscheine für Einsteiger, können switching costs gelindert werden. Manche Unternehmen übernehmen auch selbst die switching costs der Anwender, zum Beispiel indem sie Anwendern für die Rückgabe eines alten Gerätes Rabatt auf ein neues geben. Zweitens können Anwender durch switching costs vom Verlassen des eigenen Standards abgehalten werden. Durch Investitionen kann ein Unternehmen eine installed base von Anwendern aufbauen, die durch switching costs an den Standard gebunden sind. Die in die installed base getätigten Investitionen können dann auf unterschiedliche Arten wieder zurückgewonnen werden. Eine Möglichkeit ist der Verkauf von Komplementärgütern an die Anwender der installed base. Bei manchen Technologien, wie zum Beispiel Druckern, müssen regelmäßig kompatible Verbrauchsgüter (Tintenpatronen, Toner...) gekauft werden. Für diese Verbrauchsgüter können höhere Preise verlangt werden, wenn Anwender durch switching costs, wie den Kaufpreis eines neuen Druckers, an die Technologie gebunden sind. Allerdings besteht auch die Gefahr, dass Drucker zuerst billig abgegeben werden, um die installed base zu vergrößern, und dann andere Anbieter die an den Standard gebundenen Kunden billiger bedienen, zum Beispiel indem sie Nachfülltinte für die Patronen anbieten.

Eine weitere Variante, wie sich Investitionen in eine installed base rechnen, ergibt sich, wenn der Lock-In länger als einen Produktlebenszyklus anhält. Denn dann wird durch eine Investition in der Anfangsphase der Kauf mehrerer Produktgenerationen gesichert. Wird zum Beispiel in einem Unternehmen SAP R/3 eingeführt, so wird das Unternehmen mit ziemlicher Sicherheit

¹Im Gegensatz dazu tritt, wie im Kapitel 4 besprochen, sehr wohl kollektiver Lock-In durch die Bedeutung von Netzwerkexternalitäten auf.

nach Ablauf der ersten Lizenz eine weitere SAP-Lizenz erwerben, da es durch die hohen switching costs, die durch eine teure SAP-Einführung entstehen, an den Standard SAP gebunden ist.

Für Investitionen in die installed base ist es von entscheidender Bedeutung bereits an den Standard gebundene User von neuen, „freien“ Usern unterscheiden zu können. Denn nur so kann effektiv in den Aufbau der installed base investiert werden, ohne damit gleichzeitig schon gebundene User zu „beschenken“. Ideal ist es, wenn das Produkt ein aus Komplementärgütern bestehendes System ist und eines der Güter für die Benutzung des Systems unbedingt notwendig ist und eine lange Lebensdauer hat. Das System Wertkartenhandy besteht zum Beispiel aus einem Mobiltelefon und komplementären Gesprächsguthaben. Ein Mobiltelefon hat eine lange Lebensdauer und ohne es ist das System wertlos. Neue Anwender können also daran erkannt werden, dass sie ein Mobiltelefon kaufen und eine neue Handynummer beantragen. Deshalb werden Mobiltelefone von Mobilfunkbetreibern gesponsert, um von den gebundenen Kunden über den Verkauf von Wertkarten zu profitieren. Es besteht keine Gefahr bereits gebundene User nochmals zu sponsern, da sich diese kein neues Handy kaufen.

Ein weiterer wichtiger Aspekt lässt sich an Angeboten für vertraglich gebundene Handyuser demonstrieren. Es sollen nicht nur ungebundene neue User, sondern auch alte User mit niedrigen switching costs erkannt werden können. Kunden bekommen, wenn sie mit einem Mobilfunkbetreiber einen neuen Vertrag über zwölf Monate abschließen, vom Mobilfunkbetreiber einen Rabatt auf ihr neu gekauftes Handy. Nach diesen zwölf Monaten sinken die switching costs mit dem Wert des Handys. Wenn das Handy ersetzt werden soll, sind die switching costs speziell niedrig, denn wenn ein neuer Vertrag mit einem anderen Anbieter abgeschlossen wird, bekommt der Anwender ein neues Handy zum günstigen Erstanmeldungspreis. Deshalb bietet zum Beispiel T-Mobile dem Kunden die Möglichkeit sich nach 12 Monaten ein weiteres Handy zum Erstanmeldungspreis zu kaufen, wenn er sich nochmals 12 Monate bindet.

Bei Investitionen in ein Netzwerk muss immer unterschieden werden, ob damit excess inertia verhindert (wie in Kapitel 4.3 beschrieben) oder eine installed-base mit switching costs aufgebaut werden soll. Geht es um die Netzwerkeexternalitäten, so sind die Investitionen vor allem bis zum Erreichen der nötigen kritische Masse entscheidend. Danach beginnt das Netzwerk von selbst zu wachsen. Für die Überwindung bestehender oder für den Aufbau neuer switching costs gibt es hingegen keine Schwelle, ab der Investitionen nicht mehr notwendig sind. Dafür ist eine schnelle Expansion des Netzwerks weniger wichtig, da das Produkt auch für wenige Anwender einen Nutzen hat. Oft treten switching costs und Netzwerkeexternalitäten aber auch gemeinsam auf und Investitionen in die installed base erfüllen beide Zwecke.

Bei der Markteinführung des Standards spielen also diverse Formen von sponsorship eine Rolle. Doch switching costs müssen schon davor, in der Entwicklungsphase des Standards, beachtet werden. Im Idealfall wird ein Standard entwickelt, der für neue Anwender geringe, aber für den Standard verwendende Anwender hohe switching costs aufweist. Ersteres hilft User anderer Standards zum Umstieg auf den eigenen Standard zu bewegen. Ein bereits mehrfach zitiertes Beispiel war die Strategie Borlands, ihr neues Programm Quattro Pro ähnlich zum Marktführer Lotus 1-2-3 zu gestalten. Eine Importfunktion für Files des alten Standards vernichtet den Großteil der switching costs durch aufgebaute Informationen. Ein ähnliches Benutzerinterface verringert den Lernaufwand für eine neue Technologie. Diese Maßnahmen der Kompatibilität müssen aber oft gegen die möglicherweise bessere Performance einer von Grund auf neuen Technologie abgewogen werden.

Durch hohe switching costs für Anwender des eigenen Standards können diese Anwender länger gebunden und Investitionen in die installed base länger genutzt werden. Ziel ist es den

Standard so zu gestalten, dass Anwender ihre eigenen switching costs erhöhen. Beim Rezeptstandard wäre das zum Beispiel möglich, indem Anwendern der Aufbau einer großen Rezeptdatenbank erleichtert wird. Mit den Küchengeräten könnte zum Beispiel eine Gratis-Software zum einfachen Eingeben von Rezepten mitgeliefert werden oder Küchengeräte könnten eine Funktion haben während des Kochens die einzelnen Schritte aufzuzeichnen und abzuspeichern. Auch ein Online-Wettbewerb, wer einer Community die meisten und besten Rezepte zur Verfügung gestellt hat, wäre denkbar.

Je mehr Konfigurationsmöglichkeiten es gibt und je persönlicher ein Produkt abgestimmt werden kann, umso mehr werden Anwender ihre eigenen switching costs erhöhen. Beispiele für solche switching costs sind Playlists bei Musikprogrammen, Bookmarks in Webbrowsern und Makros in Office-Programmen.

Auch durch den Kauf mehrerer kompatibler langlebiger Geräte erhöhen Anwender ihre eigenen switching costs. Zum Beispiel bei Videospiele wird durch den Verkauf von Zubehör, wie etwa besserer Joysticks, nicht nur der Wert der installed base ausgenutzt, sondern auch der Lock-In der Anwender vergrößert.

Zu diesen Beispielen von gesenkten beziehungsweise erhöhten switching costs seien allerdings noch einige Einschränkungen genannt. Erstens sind Konsumenten natürlich vorsichtig, wenn sie Lock-In in einen Standard befürchten. Spielt Angst vor Lock-In eine große Rolle bei der Kaufentscheidung, kann es sinnvoll sein die switching costs für Konsumenten gering zu halten oder den Standard offen für andere Unternehmen zu gestalten.

Zweitens werden durch eine Senkung jener switching costs, die Anwender bei einem Wechsel zum neuen Standard überwinden müssen, oft auch die switching costs gesenkt, die für Anwender anfallen, wenn sie diesen neuen Standard in Zukunft verlassen wollen. Wird zum Beispiel für ein neues Dokumentformat ein zu MsWord-Dateien ähnliches Format gewählt, um eine vollständige Konvertierung von MsWord-Dateien zu ermöglichen, so wird damit auch die Konvertierung in umgekehrter Richtung erleichtert. Es ermöglicht Microsoft einen Konverter von Dateien des neuen Standards zu MsWord-Dateien zu entwickeln und die switching costs in beide Richtungen werden gesenkt.

Drittens muss beachtet werden, dass die switching costs zur Konkurrenz möglicherweise auch beim Update der eigenen Produkte auftreten. Gibt es zum Beispiel keine Möglichkeit die Rezepte eines Küchengerätes auszulesen, so verliert der Anwender beim Wechsel zu einem anderen Standard auf jeden Fall sämtliche Informationen und hat dadurch relativ hohe switching costs, da keine Konvertierung der Rezepte zum neuen System möglich ist. Aber genauso treten switching costs auf, wenn der Konsument zur nächsten Generation des gleichen Systems wechseln will.

Klemperer [34] untersucht eine interessante Konsequenz von switching costs auf die Kooperationsstrategie von Unternehmen. Er zeigt, dass Kunden Multi-Produkt-Unternehmen bevorzugen, wenn zwischen den Produkten des Unternehmens keine, aber zwischen Produkten unterschiedlicher Unternehmen sehr wohl switching costs auftreten. Im Falle eines Lock-Ins ist der Konsument dann zumindest auf ein Unternehmen mit vollständiger Produktpalette eingeschränkt. Die schon oben erwähnte Präferenz von Fluglinien ihre ganze Flotte von einem einzigen Anbieter zu beziehen ist ein gutes Beispiel. In diesem Markt können deshalb nur Multi-Produkt-Firmen, die alle benötigten Flugzeugtypen für eine Flotte herstellen, überleben. McDonnell-Douglas hatte keine komplette Familie an Flugzeugen und musste sich unter anderem deshalb 1996 von Boeing aufkaufen lassen[48]. Kleine Firmen müssen aus diesem Grund danach trachten für ihren Standard eine Koalition aufzubauen, die eine möglichst breite Palette an Produkten abdeckt.

Eine zweite mögliche Konsequenz dieser Überlegung ist, dass Firmen mehr Produkte anbieten, als ohne switching costs rational und umsatzoptimal wäre[34]. Denn es besteht die Chance Kunden, die eines dieser zusätzlichen Produkte erwerben, auch als Kunden für die restliche Produktpalette zu gewinnen. So könnte es im Beispiel eines Flugzeugherstellers einen überproportionalen Vorteil ergeben, wenn er seiner Produktpalette ein weiteres Flugzeug hinzufügt, das es ihm ermöglicht etwa für lokale Fluglinien eine komplette Flotte anzubieten.

Kapitel 7

Offenheit von Standards

Offenheit von Technologien, Standards und Software ist seit Jahren eines der zentralen Themen der IT-Branche, wie sich schon allein an den Diskussionen rund um Microsoft zeigen lässt. Zwei der zentralen Punkte im Verfahren gegen Microsoft drehten sich um Offenheit. Das waren einerseits die Forderungen nach einem offeneren Windows-API Standard und andererseits die Vorwürfe, Microsoft habe versucht den offenen Java-Standard zu fragmentieren.

Doch Offenheit ist ein Begriff mit vielen Gesichtern. Für Microsofts Gegner ist das Windows-API das klassische Beispiel eines proprietären Standards, für Microsoft hingegen ein für alle offenes Interface. Deshalb werden wir im Kapitel 7.1 drei unterschiedliche Formen von Offenheit herausarbeiten und ihre Konsequenzen diskutieren. Der Schwerpunkt liegt jedoch auf der Analyse von Fallbeispielen von mehr oder weniger offenen Standards in Kapitel 7.2. Denn die Offenheit einer Technologie wird durch viele unterschiedliche und fein abstimmbare Aspekte bestimmt, so dass die Betrachtung einzelner Fälle aufschlussreicher als deren Verallgemeinerung ist.

7.1 Formen der Offenheit

Zunächst entwickeln wir drei prinzipielle Formen oder Bedeutungen von „Offenheit“ definiert. Jede dieser drei Formen kann man sich als Achse vorstellen, deren beiden Extremwerte „offen“ und „proprietär“ sind. Zwischen den beiden Extremen gibt es, wie wir sehen werden, zahlreiche Abstufungen. Die Abbildungen 7.1 bis 7.3 veranschaulichen diese Achsen. Sie geben jeweils die Vorteile eines auf dieser Achse offenen und proprietären Standards an. Die Pfeile zeigen, welche Faktoren einen Standard auf dieser Skala in Richtung offen oder proprietär beeinflussen.

Unsere drei Skalen der Offenheit lassen sich durch die folgenden Fragen repräsentieren:

1. Können alle Unternehmen und Anwender den Standard gleich gut verwenden wie das Unternehmen, das ihn eingeführt hat?
2. Können alle Unternehmen und Anwender die Spezifikation des Standards mitbestimmen?
3. Lässt der Standard für Anwender einen Spielraum, um ihn an ihre eigenen Bedürfnisse anzupassen?

Ein System ist offen im Sinne der ersten Definition, wenn der Standard vollständig bekannt und ausreichend dokumentiert ist, so dass das Unternehmen oder die Kooperation von Unternehmen, die ihn verwalten, keine entscheidenden Vorteile bei der **Implementation** des Standards haben. Manche der im folgenden aufgezählten Maßnahmen zur Reduktion der Offenheit sind uns schon im Kapitel 5.5 beim Verhindern von free-ridership begegnet.

Durch die folgenden Maßnahmen verschieben Unternehmen den Grad der Offenheit ihrer Standards auf unserer ersten Skala in Richtung „proprietär“:

- Wenn eine Firma Rechte für den Standard kontrolliert, können andere Firmen zu Lizenzgebühren gezwungen oder ganz vom Standard ausgeschlossen werden.
- Wenn in die Dokumentation eines Standards nicht das volle Know-How einfließt, das das entwickelnde Unternehmen über den Standard besitzt, oder gar Teile des Standards nicht dokumentiert werden, so sichert sich das Unternehmen einen Wettbewerbsvorteil gegenüber den Konkurrenten.
- Einen ähnlichen Vorteil sichert sich ein Unternehmen, wenn Spezifikationen zuerst an die eigenen Entwicklungsabteilungen weitergegeben werden. Speziell beim Auftreten starker Netzwerkexternalitäten kann ein dadurch entstehender first-mover advantage entscheidend sein.

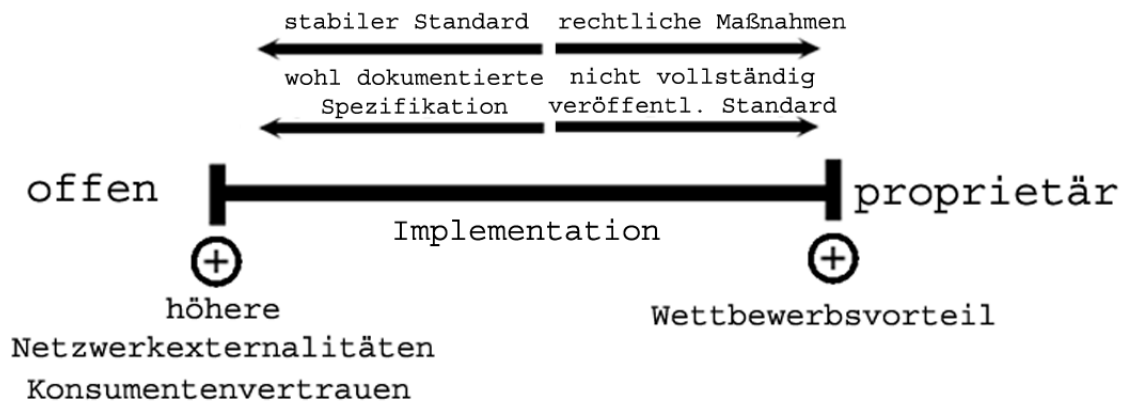


Abbildung 7.1: 1. Skala - Offene Verwendbarkeit eines Standards

- Die Offenheit wird auch durch instabile Standards, die sich oft verändern, eingeschränkt, was vor allem in Kombination mit dem vorigen Punkt, der zeitlich verzögerten Veröffentlichung, einen deutlichen Vorteil schaffen kann. Bei Produkten, die ohnehin regelmäßig upgedatet werden müssen, wie z.B. Virenschannern, ist eine häufige Veränderung des Standards besonders leicht möglich.

Ein offener, für alle gleich gut verwendbarer Standard schafft für alle Beteiligten einen Vorteil durch höhere Netzwerkexternalitäten. Er wird von mehr Firmen angewandt als eine proprietärere Variante, die ja eben zum Ziel hat Konkurrenten die Nutzung des Standards zu erschweren. Außerdem wird durch die Offenheit das Vertrauen von Konsumenten in den Standard erhöht, da die Gefahr eines Lock-In reduziert wird, wenn der Standard von mehreren Unternehmen angeboten werden kann.

Durch einen proprietären Standard kann ein Unternehmen hingegen verschiedene Formen von Wettbewerbsvorteilen erreichen. Wie in der obigen Aufzählung beschrieben, können dadurch Konkurrenten ausgeschlossen, überlegene Produkte erzeugt oder Lizenzgebühren eingenommen werden.

Die zweite Form der Offenheit reicht schon weiter, sie beschreibt das Ausmaß, in dem fremde Unternehmen an der **Entwicklung** des Standards oder der Technologie teilnehmen können. Die radikalste Form der Offenheit sind open-source-Projekte. Bei solchen Projekten wird nicht nur die Interface-Spezifikation sondern der gesamte Quelltext einer Software veröffentlicht. Darüber hinaus kann meist jeder, freiwillig und unentgeltlich, an der Entwicklung der Software mitarbeiten. Sowohl Anwender als auch Unternehmen können also erstens die Spezifikation des Standards mitbestimmen und zweitens nach der Fertigstellung weiterentwickeln und an ihre persönlichen Bedürfnisse anpassen. Zwei der bekanntesten Beispiele sind das Betriebssystem Linux und der Browser Netscape, der seit 1998 vom mozilla.org Projekt im open-source Verfahren entwickelt und im Kapitel 7.2 noch ausführlich beschrieben wird.

Eine andere Möglichkeit ist, die Spezifikation des Standards an eine neutrale Organisation zu übergeben, die dann einen offenen Entwicklungsprozess koordiniert, an dem alle interessierten Unternehmen teilnehmen können. Ein Beispiel ist etwa die Verwaltung des Unix-Standards, die von Novell 1994 der X/Open Group übertragen wurde (für Details s.a. Kapitel 7.2).

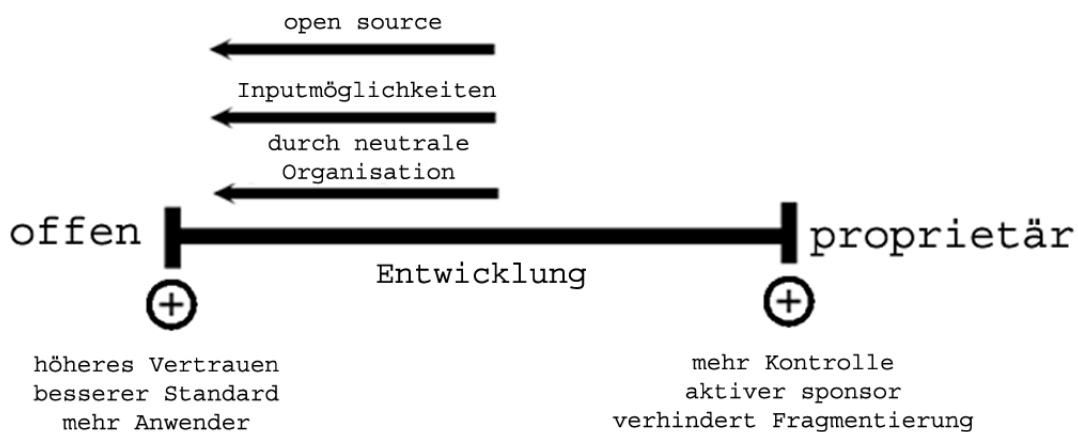


Abbildung 7.2: 2. Skala - Offene Entwicklung eines Standards

Schon deutlich weiter in Richtung proprietärer Standard liegt ein Entwicklungsprozess, der von dem Unternehmen mit den Rechten am Standard kontrolliert wird, in den aber andere Unternehmen Input einbringen können. Das kann von einer Beteiligung in Beratungs- oder Entscheidungsgremien bis zur bloßen Meldung von Software-Bugs reichen. Ein Beispiel ist der Java Community Process zur Entwicklung der Java-Spezifikation, der ebenfalls im nächsten Kapitel genauer behandelt wird.

Die Vorteile einer offenen Standardentwicklung sind wieder höhere Akzeptanz und höheres Vertrauen in den Standard. Für Firmen, die nicht die Rechte an einem Standard besitzen, wird durch eine Partizipation einerseits die Wahrscheinlichkeit des Lock-In weiter reduziert, aber vor allem auch die Gefahr beseitigt, dass sich der Standard in eine unvorteilhafte Richtung entwickelt. Außerdem werden mehr Unternehmen den Standard adaptieren, da sie ihn besser auf ihre Bedürfnisse abstimmen können. Eine offene Standardentwicklung kann auch zu einer verbesserten Qualität des Standards führen, weil mehr und vor allem spezialisierteres Know-How zur Verfügung steht. So sind zum Beispiel in den Gruppen für die Entwicklung der Java Micro Edition für mobile Geräte Mobiltelefonhersteller wie Nokia oder Motorola besonders stark vertreten[3].

Diesen Vorteilen stehen aber auch einige Nachteile gegenüber. Ganz direkt verliert ein Unternehmen, das die Entwicklung eines Standards für andere Unternehmen öffnet, Kontrolle über die Technologie, Imagevorteile und unter Umständen Lizenznahmen. Durch diese Nachteile wird die Technologie mehr zu einem public good, die Vorteile, die das einführende Unternehmen für sich allein gehabt hat, sinken. Dadurch verliert das Unternehmen den Anreiz die Technologie zu sponsern und in die Netzwerkgröße zu investieren. Offenen Standards fehlt daher manchmal ein Sponsor, der die Entwicklung der Technologie vorantreibt.

Ein fehlender Sponsor kann auch zur Fragmentierung eines Standards führen. Eine Fragmentierung entsteht, wenn unterschiedliche Ausprägungen eines Standards entstehen, die miteinander nicht mehr voll kompatibel sind. Dadurch sinken die Netzwerkexternalitäten für diesen Standard, da aus einem großen mehrere kleine Netzwerke werden. Das klassische Beispiel für eine solche Fragmentierung auf Grund eines fehlenden Sponsors ist der Unix-Standard, dessen Geschichte ebenfalls im Kapitel 7.2 genauer beschrieben wird. Fragmentierung ist auch eine der großen Gefahren von open-source Projekten. Denn Weiterentwicklungen eines open-source Programms, die nicht in der zentralen Version des Programms nachgezogen werden, können zu

inkompatiblen Fragmenten führen. Doch es gibt eine Kraft, die der Fragmentierung entgegenwirkt: Netzwerkexternalitäten. Unternehmen, die ein open-source Programm weiterentwickeln, wollen meist nicht auf die Netzwerkexternalitäten des bestehenden Netzwerks verzichten und werden daher danach trachten ihre neue Version kompatibel zum bestehenden Standard zu gestalten. Versucht ein Unternehmen eine zum Standard inkompatible Version des Programms zu verbreiten, so muss es selbst ein neues Netzwerk aufbauen und die excess inertia überwinden. Auch wenn Netzwerkexternalitäten die Wahrscheinlichkeit einer Fragmentierung senken, zeigt doch die Geschichte des HTML-Standards, dass sie die Fragmentierung eines Standards nicht ganz verhindern können. Die Browser Microsoft Internet Explorer und Netscape Navigator stellen HTML-Seiten beide nicht vollkommen standardkonform und vor allem unterschiedlich dar, sodass Webentwickler dazu gezwungen werden ihre Webseiten auf beiden Systemen zu testen. Diese Fragmentierung entstand vor allem, weil beide Browser bereits eine ausreichende installed-base aufgebaut hatten und nicht mehr mit excess inertia kämpfen mussten. Im Gegensatz dazu bietet Opera, ein Browser mit relativ kleiner installed base, sogar die Möglichkeit auszuwählen als welcher anderer Browser er sich ausgeben soll. Er versucht also beide Fragmente zu unterstützen, um möglichst große Netzwerkexternalitäten zu generieren. An der HTML-Fragmentierung lässt sich auch erkennen, dass selbst unabsichtliche Fragmentierung oder Fragmentierung durch technische Zwänge eine Gefahr darstellen. So unterscheidet sich nämlich auch die Interpretation von HTML Seiten zwischen verschiedenen Versionen des selben Browsers¹ und dass Microsoft oder Netscape freiwillig auf Netzwerkexternalitäten verzichten, indem sie verschiedene Versionen ihres eigenen Browsers nicht vollständig kompatibel gestalten erscheint unwahrscheinlich.

Mit dem Verlust von Kontrolle über die Technologie geht oft auch eine Verlangsamung der Standardentwicklung und der Reaktionszeit für Produktänderungen einher. Denn die Abläufe einer offenen Standardisierung mit einer großen Anzahl von beteiligten Organisationen sind schwerfälliger als der Entscheidungsapparat einer einzelnen Firma.

Die dritte Form der Offenheit betrifft die **Anpassbarkeit** der Standard Spezifikation durch den End-User. Eine solche Anpassbarkeit kann auf zwei Arten erreicht werden. Erstens durch eine sehr umfangreiche Spezifikation des Standards, die praktisch die Wünsche aller User miteinbezieht, so dass jeder User die für seine Zwecke relevanten Teile des Standards auswählen kann. Ein Beispiel dafür ist der EDIFACT-Standard für den elektronischen Datenaustausch (EDI²) zwischen Unternehmen (siehe Kapitel 7.2). Zweitens können Usern Möglichkeiten zur Anpassung des Standards gegeben werden, etwa frei belegbare Speicherbereiche auf einer Smartcard.

Der Vorteil einer in diesem Sinne offenen Technologie ist offensichtlich die gesteigerte Attraktivität für User. Die Zahl der mit dieser Technologie lösbaren Probleme wird erhöht, dem User stehen flexiblere Instrumente zur Verfügung. Doch die große Gefahr besteht wieder in der Fragmentierung. Wird eine sehr umfangreiche Spezifikation gewählt, so werden Programme, die den Standard verwenden vielleicht nicht die gesamte Spezifikation unterstützen können, wie das beim EDIFACT-Standard passiert ist. Unterschiedliche Programme verwenden daher unterschiedliche Teile des Standards und sind nicht mehr kompatibel. Auch vom User frei definierbare Elemente können leicht zur Fragmentierung führen, vor allem wenn sie in für die Kompatibilität wichtigen Bereichen verwendet werden. Sinnvoll sind frei verwendbare Elemente vor allem dann, wenn der Standard in Teile, die für die Kompatibilität wichtig sind, und in Teile, für die umfangreiche Kompatibilität keine entscheidende Rolle spielt, eingeteilt werden kann. Das lässt sich am Beispiel des Quick-Standards, der in Österreich für bargeldlose Bezahlung mit

¹Z.B. zwischen IE 4 und IE 5, der Unterschied ist aber wesentlich geringer als der zwischen Netscape Communicator und Internet Explorer.

²Electronic Data Interchange

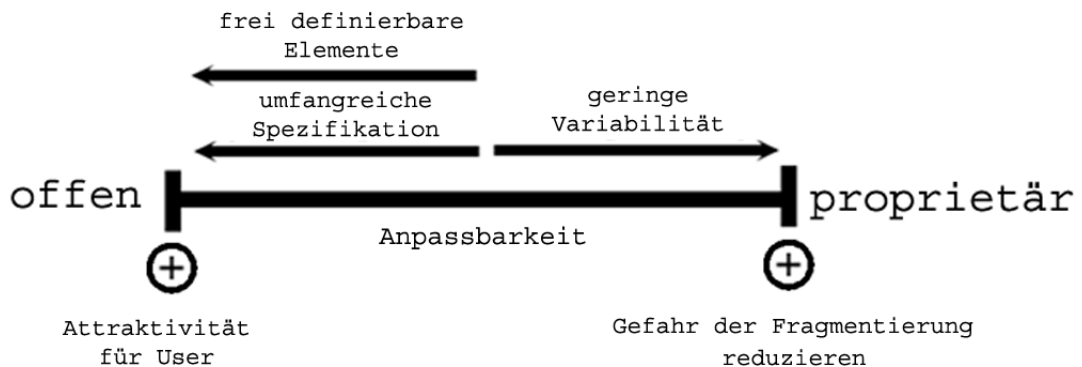


Abbildung 7.3: 3. Skala - Anpassbarkeit eines Standards

Chipkarten verwendet wird, gut veranschaulichen. Auf diesen Chipkarten können elektronisch Geldbeträge mit einem Wert von bis zu 400 EUR gespeichert werden. Zusätzlich enthält der Chip noch Felder, die im Rahmen eines so genannten „Affinity“-Programms, von Unternehmen zur Speicherung von Kundendaten verwendet werden können. So speichern zum Beispiel die Ketten BIPA und Merkur Informationen über ihre Stammkundenprogramme auf der Karte und auf den Universitäten Linz und WU Wien werden solche Karten als Studentenausweise eingesetzt[2]. Solange diese freien aber proprietären Informationen nur für die Transaktion mit einer bestimmten Firma verwendet werden, bietet diese Lösung vor allem hohen Komfort für die Anwender, birgt aber nicht die Gefahr einer Fragmentierung. Doch wenn etwa die gespeicherten Studentenausweis-Daten auch von anderen Stellen zur Identifikation akzeptiert werden, verbreitet sich ein proprietärer Standard der Linzer Universität, der sich zum Beispiel vom Standard einer geplanten „Bürgerkarte“ unterscheiden könnte. In diesem Szenario entstehen also zwei konkurrierende Standards zur Identifikation mit Quick-Karten, der Standard wurde fragmentiert. Unternehmen oder Behörden, die elektronische Identifikation akzeptieren, müssen entweder beide Standards unterstützen oder geringere Netzwerkexternalitäten in Kauf nehmen, da sie nur einen Teil der Karten verstehen können.

Bei manchen Standards, wie zum Beispiel Programmiersprachen, ist Flexibilität allerdings nicht optional. Denn es ist ja gerade der Sinn einer Programmiersprache Anwendern das Entwickeln eigener Funktionen mit eigenen Schnittstellen zu ermöglichen. Die dadurch entwickelten proprietären Programmbibliotheken oder Klassen stehen nur einem Teil der Programmierer zur Verfügung. Wie das zu Fragmentierung führen kann wird anhand von Java im nächsten Kapitel besprochen.

7.2 Offenheit an Fallbeispielen

Nachdem nun die grundlegenden Merkmale der drei Offenheits-Formen beschrieben wurden, beleuchtet dieses Kapitel die Anwendung unterschiedlicher Offenheits-Strategien in der Praxis. Dazu analysieren wir die folgenden Beispiele:

- Mozilla, ein open-source Browser
- Java, eine offen entwickelte und veröffentlichte Programmiersprache

- Windows, ein proprietäres Betriebssystem
- Unix, ein Betriebssystem, das viele Formen von Offenheit durchlaufen hat
- EDIFACT, ein offen entwickelter und implementierter EDI-Standard

Im Jänner 1998 gab Netscape dem so genannten „browser war“ zwischen Internet Explorer und Netscape Communicator eine neue Wendung, indem sie den gesamten Source Code ihres Browsers Netscape Communicator 5.0 veröffentlichten[37]. Wenig später wurde Mozilla.org, eine Organisation zur Steuerung des open-source Entwicklungsprozesses, gegründet und dem öffentlich verfügbaren Quelltext der Name Mozilla gegeben[38]. Die Netscape Public License (NPL) gestattet es sowohl privaten Anwendern als auch Unternehmen den Mozilla-Quelltext, als Ganzes oder Teile davon, für ihre eigenen Produkte zu verwenden. Netscape selbst verwendet den Mozilla-Quelltext zum Beispiel als Grundlage für die Versionen 6 und 7 des Communicator-Browsers und erweitert ihn noch um zusätzliche Funktionen, die nur im Communicator, aber nicht in Mozilla zur Verfügung stehen. Die Weiterentwicklung von Mozilla erfolgt einerseits durch ein ungefähr 20-köpfiges Team von Mozilla.org und andererseits durch freiwillige Inputs aus der Internet Community. Die Koordination und vor allem die endgültige Entscheidung, welche Funktionen dem Mozilla-Standard hinzugefügt werden, behält sich Mozilla.org allerdings selbst vor. Neben den aus Mozilla entwickelten Netscape Versionen wurde im Juni ein voll funktionsfähiger Browser Mozilla 1.0 veröffentlicht.

Bewerten wir den Mozilla-Standard auf allen drei Skalen der Offenheit, so ist er bezüglich aller drei Aspekte sehr offen. Der erste Aspekt, die Verwendbarkeit der Standard-Definition für andere Unternehmen, könnte nicht größer sein, denn es werden alle Details des Projektes mit täglicher Aktualität und wohl dokumentiert auf mozilla.org veröffentlicht. Bezüglich der zweiten Skala, der Beteiligung an der Entwicklung des Standards, ist Mozilla auf zweifache Weise offen. Einerseits kann sich jeder an der Entwicklung der Spezifikation beteiligen und andererseits kann jeder den Mozilla-Standard nach der Veröffentlichung noch an seine eigenen Bedürfnisse anpassen. Die einzige Einschränkung, dass letztendlich die Mozilla-Organisation entscheidet, was zum Standard wird, ist wohl unerlässlich um eine konsistente Entwicklung des Standards zu sichern. Auf der dritten Skala, der Anpassbarkeit durch den User, ist Mozilla zumindest theoretisch sehr offen, auch wenn in der Praxis das Ändern des eigenen Browsers auf Quelltextebene nicht zu den dringlichsten Bedürfnissen der meisten Anwender zählen dürfte.

Warum hat Netscape nun diese radikale Strategie der Offenheit gewählt? Was versprach man sich davon den Großteil des geistigen Eigentums zu verschenken?

Erstens hatte Netscape durch diesen Schachzug nicht allzu viel zu verlieren. Die Einnahmen aus Browser-Lizenzen machten im letzten Quartal des Jahres 1997 nur mehr 13% des Gesamtumsatzes aus, den Großteil nahm Netscape durch Verkäufe anderer Technologien, wie seinem Webserver, und durch Werbung auf seiner Website ein[37]. Der Browser war und ist neben den Lizeineinnahmen vor allem aus drei weiteren Gründen von Bedeutung. Er ist erstens wichtig für Netscapes Reputation. Zweitens verkauft Netscape Links in den Lesezeichen und in seinen Symbolleisten an andere Unternehmen. Drittens erhöht Netscape die Anzahl der Besucher seiner Homepage, indem zahlreiche Verbindungen zu der Homepage im Browser voreingestellt sind und im inkludierten Mailprogramm Teile der Homepage permanent angezeigt werden. Solange der Begriff Mozilla mit dem Unternehmen Netscape verbunden bleibt, verliert Netscape nicht die mit dem Browser verbundene Reputation. Außerdem veröffentlicht Netscape ja weiterhin seinen eigenen Browser, den Communicator, eine erweiterte und verbesserte Version des Mozilla-Standards. Solange der Communicator nicht allzu viel Marktanteil an den Mozilla-Browser verliert, können auch die Werbeeinnahmen aus dem Communicator bewahrt werden.

Zweitens konnte Netscape an Qualität gewinnen und Kosten senken, indem sie einen Teil des Entwicklungsaufwand zu der Internet Community verlagerten. Netscape baute darauf, dass zahlreiche Entwickler kostenlos ihre Arbeit zur Verfügung stellen würden, motiviert durch Anerkennung in der Community, das Bedürfnis gegen Microsofts Vorherrschaft anzutreten und persönliches Interesse an neuen Features des Browsers. Auch wenn heute ein beträchtlicher Teil des Inputs aus der Internet Community kommt, wird ein wichtiger Teil der Entwicklungsarbeit noch immer von Netscape getragen[36].

Der dritte und wichtigste Punkt ist, dass das open-source Modell Netscape eine extreme Diversifizierung ermöglicht, der Microsofts Internet Explorer als relativ unflexibler Koloss gegenüber steht. Diese Diversifizierung wird dadurch ermöglicht, dass jeder den Mozilla-Code an seine eigenen Bedürfnisse anpassen kann. So können Vertreter von Minderheiten Mozilla-Versionen in ihren Sprachen erstellen und Blindenorganisationen können Mozilla für Blinde adaptieren. Unternehmen können den Mozilla-Source-Code an ihre eigenen Bedürfnisse anpassen, zum Beispiel um firmeninterne Backend-Systeme in das Firmen-Intranet zu integrieren. Weiters könnten Unternehmen ihren Kunden für ihre Produkte adaptierte Versionen von Mozilla zur Verfügung stellen, etwa mit einem adaptierten Menü für eigene e-Commerce Komponenten. Die Anpassbarkeit des Internet Explorers ist dagegen extrem eingeschränkt, Unternehmen sind abhängig von den von Microsoft zu bestimmten Zeitpunkten veröffentlichten Versionen und können den Browser nur geringfügig an ihre Bedürfnisse anpassen.

Die große Gefahr, die von Mozillas Potential zur Diversifikation ausgeht, ist eine Fragmentierung des Standards. Wie schon erwähnt spielt im Browsermarkt die Größe des Netzwerks eines Browsers vor allem deshalb eine Rolle, weil unterschiedliche Browser Webseiten unterschiedlich interpretieren. Da die meisten Webseiten für den Browser mit der größten Verbreitung optimiert werden, ist es für Anwender vorteilhaft dem größeren Netzwerk beizutreten. Sind nun die von der Mozilla-Basis weiterentwickelten Browser nicht vollständig mit dem Mozilla-Standard kompatibel, so erzeugen sie auch keine Netzwerkexternalitäten für Mozilla und Netscape.

Wie kann Netscapes Strategie den source-code zu veröffentlichen vier Jahre später beurteilt werden? Der große Erfolg für Mozilla und Netscape ist bis heute jedenfalls ausgeblieben. Der Marktanteil des Netscape Browsers lag im August 2002 nur noch bei 3,4% im Vergleich zu 13% im Jahr davor[6]. Grund für diesen Verlust an Marktanteilen war die lange Entwicklungsdauer des Browsers. Die erste auf Mozilla basierende Vollversion Netscape 6.0 wurde im November 2000 veröffentlicht, also erst zweieinhalb Jahre nach der Gründung von mozilla.org. Netscape 6.0 wurde jedoch von den meisten Kritikern als instabil, speicheraufwendig und dem Internet Explorer klar unterlegen bezeichnet. Version 1.0 von Mozilla wurde erst im Juni 2002 veröffentlicht, ist aber im Gegensatz zu Netscape 6.0 dem aktuellen Internet Explorer performancemäßig zumindest gleichwertig. Mozillas Problem ist, dass Microsoft in den vier Jahren seit 1998 in aller Ruhe seinen Internet Explorer verbreiten konnte, den Browsermarkt mittlerweile mit 96% Marktanteil dominiert und damit gegenüber Mozilla und Netscape einen erheblichen Vorteil aus Netzwerkexternalitäten genießt. Doch Anhänger von Mozilla sehen einige Hoffnungsschimmer am Horizont. Erstens bietet Microsoft den Internet Explorer nur für Windows und Macintosh an, im Nischenmarkt anderer Betriebssysteme wird noch immer hauptsächlich Netscape verwendet. Zweitens, und schon bedeutender, überlegt AOL, der führende amerikanische Internet-Provider und nebenbei Eigentümer von Netscape, für seine Internet-Zugangssoftware vom Internet Explorer zu Mozilla zu wechseln. In der AOL-Tochter Compuserve läuft bereits ein Mozilla-Pilotprojekt. Mozilla könnte damit auf einen Schlag 30 Millionen neue Anwender gewinnen. Drittens hat Mozilla das Potential mehr als nur ein Internet-Browser zu sein, nämlich eine Basis für viele andere Userschnittstellen: zum Beispiel entwickeln Intel, Nokia und WorldGate neue digitale

Fernsehlösungen, die Mozilla Elemente enthalten[15]. Zahlreiche andere Unternehmen, wie Sun, Red Hat oder IBM verwenden Mozillas „rendering engine“ Gecko in ihren Produkten[39].

Realistisch betrachtet hat Netscape diese Phase des browser wars, vor allem auf Grund des zeitlichen Rückstandes, verloren, aber durchaus Chancen mit Mozilla im Bereich angepasster Internetbrowser Marktanteile zurückzuerobern.

Unser zweites Beispiel ist die von Sun 1995 eingeführte Programmiersprache Java. Javas wichtigstes Motto lautet „write once, run everywhere“, in Java geschriebene Programme laufen also, ohne verändert zu werden, auf allen Computerplattformen mit den gleichen Ergebnissen. Um das zu erreichen wird für jede Plattform eine eigene sogenannte „Java Virtual Machine“(JVM) zur Verfügung gestellt, die Java Programme in die Sprache der jeweiligen Plattform übersetzt. Wie schon in Kapitel 5 beschrieben, steht die gesamte Java-Umgebung allen Anwendern kostenlos zur Verfügung. Nur Unternehmen, die in ihren Produkten (z.B. Internet-Browsern) eine JVM verwenden wollen, müssen Lizenzgebühren bezahlen. Sun verdient an Java durch diese Lizenzgebühren, Consultingleistungen und den Verkauf weiterer mit Java verwandter Produkte.

Offenheit auf der ersten Skala ist ein erklärtes Ziel von Java. Der Standard soll für jeden verwendbar sein, die Spezifikation ist vollständig veröffentlicht und ausführlich dokumentiert. Sun verzichtet also auf Einnahmen aus einer Lizenzierung des Standards, um Netzwerkexternalitäten zu schaffen und den Standard rasch zu verbreiten.

Wie offen die Erstellung der Java-Spezifikation ist, ist schon weniger eindeutig zu beurteilen. Partner von Sun, die wesentliche Teile ihres Geschäftes auf Java-Technologien aufbauen, wie zum Beispiel IBM, fordern regelmäßig, dass Sun die Weiterentwicklung der Java-Technologie an eine unabhängige Organisation übergibt. Sie sehen es als Gefahr, dass Sun als einer ihrer Konkurrenten die Kontrolle über eine für sie wichtige Technologie hat und den Java-Standard in eine für sie ungünstige Richtung lenken könnte. So könnte Sun zum Beispiel für IBM wichtige Elemente von Java verändern oder auflassen.

Sun hingegen, ist nicht bereit die Kontrolle über Java ganz aufzugeben. Das würde in erster Konsequenz einmal zu einem Verlust von Einnahmen führen. Sun könnte keine Lizenzgebühren mehr einheben und ihr Know-How Vorsprung und damit Einnahmen aus Consultingleistungen würden schrumpfen. Außerdem fürchtet Sun eine Fragmentierung des Java-Standards, wenn sie nicht mehr als starker Sponsor hinter Java stehen, in das Java-Netzwerk investieren und auf strikte Einhaltung der Kompatibilitätskriterien achten. Um sein Bekenntnis zu einer offenen Java-Spezifikation zu demonstrieren, aber dennoch die Kontrolle über den Standard zu bewahren hat Sun den „Java Community Process“(JCP) zur Weiterentwicklung von Java gestartet. Der JCP ermöglicht es interessierten Unternehmen in für sie relevanten „working groups“ an der Spezifikation von Java mitzuarbeiten, er bietet ein formalisiertes und demokratisches Vorgehensmodell, dessen Einhaltung auf Suns Auftrag von PricewaterhouseCoopers überwacht wird[49]. Allerdings behält sich Sun bei allen Entscheidungen ein Vetorecht vor. Auch wenn die „working groups“ eigenständig die Richtung der Technologie bestimmen, liegt die letzte Entscheidung also immer noch bei Sun[3].

Suns Angst vor einer Fragmentierung des Java-Standards ist aus mehreren Gründen durchaus berechtigt. Einer davon liegt in der Offenheit auf der dritten Skala, der Anpassbarkeit durch den User. Java ist als objektorientierte Programmiersprache sehr modular aufgebaut, sodass jeder Anwender und jedes Unternehmen neue Klassen entwickeln und anderen Anwendern zur Verfügung stellen kann. Dadurch entsteht die Gefahr, dass außerhalb des JCP proprietäre Java-Klassen entwickelt werden, die zu einem defacto-Standard avancieren, aber nicht in der Standard-Auslieferung der Java-Plattform enthalten sind. Java Anwendungen, die unter Verwendung dieser Klassen geschrieben wurden, laufen dann nur noch auf Systemen, die diese Klassen

unterstützen. Damit wäre das „write once, run everywhere“-Paradigma von Java gefährdet. Eine solche Gefahr droht Java zum Beispiel von dem von IBM entwickelten Eclipse-Toolkit, das Java-Erweiterungen für Web Services enthält[3].

Eine Fragmentierung auf einer anderen Ebene hatte Microsoft schon 1997 versucht und wurde dafür von Sun verklagt. Microsoft hatte in seinem Internet Explorer 4.0 eine inkompatible Java Virtual Machine implementiert. Neben fehlender Unterstützung für die Interfaces JNI und RMI hatte Microsoft auch Methoden aus den Standard-Java packages entfernt und einige Felder hinzugefügt. Im Jahr darauf veröffentlichte Microsoft das Java Entwicklungstool „Visual J++“, das wiederum neue Befehle einführte, die nur auf der Windows-Plattform funktionierten. Diese beiden Maßnahmen hatten zur Folge, dass einerseits Java-Programme geschrieben werden konnten, die nur auf der Windows-Oberfläche funktionierten und andererseits Programme, die nicht speziell für Windows geschrieben wurden, unter Umständen nicht auf Windows ausführbar waren. Java wäre in ein Windows-kompatibles und ein nicht Windows-kompatibles Segment zerfallen, der „write once, run everywhere“-Grundsatz wäre gefallen. Was erhoffte Microsoft damit zu erreichen? Vor der Einführung von Java wurde der Großteil der Anwendungen für das Windows API geschrieben, da Windows das größte Netzwerk hat. Da es deshalb für Windows die meisten Anwendungen, gab vergrößerte sich das Netzwerk weiter. Dieses positive feedback war durch Java gefährdet, denn Entwickler konnten für alle Plattformen gleichzeitig programmieren und die Anzahl der Applikationen für nicht-Windows Betriebssysteme wäre wieder gestiegen. Wäre nun „Java für Windows“ inkompatibel zur restlichen Java-Welt, so würden Programmierer wieder beginnen ihre Anwendungen nur für Windows zu entwickeln.

Der Sun-Microsoft Prozess endete mit einem Vergleich, nach dem Microsoft \$20 Mio. an Sun bezahlen musste und ein permanentes Verbot für das Tragen der „Java-Compatible“ Trademark auferlegt bekam. Sun kündigte weiters die Lizenzvereinbarung mit Microsoft, Microsoft darf aber Java in Produkten, in denen es bereits integriert ist, für sieben Jahre weiterverwenden[45].

Eine ganz andere Strategie wendet Microsoft bei seinem eigenen Windows API Standard an. Dieser Standard ist auf der Skala der Offenheit der Entwicklung zweifellos proprietär. Nur Microsoft bestimmt, wie das Windows-API der nächsten Generation auszusehen hat. Interessanter ist schon die Frage, wie offen der Standard für die Verwendung durch Konkurrenzunternehmen ist. Da Microsoft sowohl auf der Ebene unter dem API (dem Betriebssystem) als auch auf der Ebene über dem API (Windows Applikationen) vertreten ist, ist die Verwendung durch andere Unternehmen für Microsoft ein zweiseitiges Schwert. Denn einerseits wollen sie die Netzwerkexternalitäten für die Windows Plattform erhöhen. Das würde dafür sprechen den Windows API Standard vollständig für die Verwendung durch andere Unternehmen zu öffnen, um die Entwicklung von Windows-Anwendungen zu fördern. Doch auf der anderen Seite bietet Microsoft auch Anwendungen auf der oberen Ebene an und kann sich durch ein weniger offenes API Wettbewerbsvorteile gegenüber der Konkurrenz im Anwendungsmarkt schaffen.

Auch Microsofts API-Strategie ist dementsprechend zweiseitig. Microsoft bekennt sich zwar zu dem API als offenem Standard, die Realität sieht aber etwas anders aus. 1994 veröffentlichten Schulman et al. das Buch „Undocumented Windows: A Programmers Guide to Reserved Microsoft Windows API Functions“[44], in dem er über 200 API Aufrufe beschrieb, die von Microsoft nicht dokumentiert worden waren. Es folgten weitere Vorwürfe und schließlich untersuchte das amerikanische Department of Justice in einem mehrjährigen Verfahren Microsofts Praktiken. In einem Vergleich im November 2001, der aber allgemein als Erfolg für Microsoft gesehen wird, verpflichtete sich Microsoft dazu geheime Windows API-Aufrufe zu veröffentlichen[21]. Besonders wichtig waren interne API-Befehle, die verwendet wurden um den Microsoft Internet Explorer enger ans Betriebssystem zu koppeln. Neben den geheimen API-Befehlen wurden auch

Vorwürfe laut, dass Microsoft neue API-Funktionen zuerst an interne Entwicklungsabteilungen weitergibt, bevor sie veröffentlicht werden[48] und dass Microsoft absichtlich Änderungen vornahm, die Anwendungen anderer Anbieter zum Absturz brachten. So behaupteten zum Beispiel Apple und RealNetworks, dass Microsoft das Windows API so manipulierte, dass ihre Multimediaspieler nicht mehr korrekt funktionierten[42].

Microsoft wählt also einen „Mittelweg“ für ihre Offenheitsstrategie, indem sie den Windows-API Standard zwar größtenteils veröffentlicht um Netzwerkexternalitäten zu generieren, sich aber selbst genug Information und Kontrolle sichern um einen deutlichen Wettbewerbsvorteil halten zu können.

Auch das Betriebssystem Unix dient als aufschlussreiches Beispiel in Sachen Offenheit. Unix wurde 1969 von AT&Ts „Bell Labs“ als Forschungsprojekt entwickelt. Der Quelltext konnte von akademischen Organisationen jahrelang kostenlos verwendet werden. In den 70er Jahren begannen immer mehr Firmen Unix-Betriebssysteme anzubieten. Allerdings veränderten und erweiterten sie Unix, um neue Funktionen hinzuzufügen. In den 80er Jahren boten zahlreiche Unternehmen, wie IBM, Sun, Hewlett-Packard, Silicon Graphics und Novell, unterschiedliche, inkompatible Unix-Versionen an. Das Bedürfnis sich durch Veränderung und Verbesserung des Betriebssystems zu differenzieren war größer als der Netzwerkvorteil, den sich Unternehmen durch einen einheitlichen Unix-Standard versprachen[48]. Es lässt sich gut erkennen, wie das Fehlen eines Sponsors, der die Entwicklung der Technologie kontrolliert, den Standard in Fragmente zerfallen ließ.

1986 starteten europäische Unix-Anbieter, unter anderem Bull, Nixdorf, Olivetti und Siemens, den ersten Versuch einen herstellerübergreifenden Standard zu entwickeln. Sie wollten sich zusammenschließen, um einen Wettbewerbsvorteil gegenüber japanischen und amerikanischen Billiganbietern zu erreichen. Doch die EG-Kommission verpflichtete sie aus kartellrechtlichen Gründen dazu einen offenen Standard zu wählen, von dem kein Unternehmen ausgeschlossen werden durfte. Glanz[19] analysiert diese Entscheidung mit Hilfe eines von ihm entwickelten netzwerkökonomischen Modells. Er zeigt, dass die europäische Kooperation aufgrund der Entscheidung der EG-Kommission wieder zerbrochen ist. Denn die beteiligten Unternehmen hatten bereits eine installed base von Usern ihres proprietären Unix-Standards aufgebaut. Würden sie sich nun einem offenen Standard anschließen, so würden sie ihren Vorteil gegenüber neu am Markt eintreffenden Konkurrenten verlieren. Die Unternehmen hätten zwar einen herstellerübergreifenden Standard dem proprietären vorgezogen, waren aber nicht bereit den Standard gänzlich zu öffnen und sich der verstärkten Konkurrenz auszusetzen.

So gab es auch in den darauf folgenden Jahren keinen einheitlichen Unix-Standard. Zu Beginn der 90er Jahre führte Microsoft Windows NT als Konkurrenz zu Unix ein und profitierte von dem fragmentierten Unix-Standard. 1993 gab es neue Bemühungen Unix zu vereinheitlichen. Novell erwarb die Bell Laboratories und damit die Rechte für die Unix Trademark, übergab die Rechte aber kostenlos an das X/Open Konsortium. Das Konsortium sollte jeder Firma das Recht geben ihr Produkt Unix zu nennen, solange es den von X/Open spezifizierten Kompatibilitätskriterien entsprach[48]. Auch wenn sich dieser Versuch nicht durchsetzen konnte, zeigt er, wie Novell auf einen entscheidenden Wettbewerbsvorteil verzichtete, um das gesamte Unix-Marktsegment zu beleben. Anstatt nach einem größeren Anteil vom Markt zu streben, versuchten sie den Markt zu vergrößern.

Als letztes Beispiel dient uns der UN/EDIFACT-Standard, der 1985 von der United Nations Commission for Europe (UN/ECE) gemeinsam mit der ISO entwickelt wurde, um einen automatischen Datenaustausch zwischen Unternehmen auf internationaler und branchenübergreifender Ebene gewährleisten zu können. EDIFACT definiert eine große Anzahl von standardisierten

Nachrichten, die das Computersystem eines Unternehmens an das Computersystem eines anderen Unternehmens übermitteln kann. Zum Beispiel gibt es einen Nachrichtentyp INVOICE, der die unterschiedlichen Felder einer Rechnung definiert. Rechnungen können je nach Branche, Land und Art der verkauften Güter sehr unterschiedlich aussehen, deshalb wurden die Nachrichtentypen sehr variabel gestaltet. Jede Nachricht enthält unzählige optionale Komponenten, durch die die Anzahl der Variationsmöglichkeiten so groß wird, dass nicht jedes System alle Varianten interpretieren kann. Weiters sind Nachrichten semantisch nicht vollständig, das heißt, jede Information kann auf unterschiedliche Weise dargestellt werden. Der Standard ist also offen im Sinne einer hohen Anpassbarkeit durch den Anwender. Doch diese Offenheit hat zu einer Fragmentierung des EDIFACT-Standards geführt. Um die Semantik von EDIFACT-Nachrichten zu vereinheitlichen, begannen Interessengemeinschaften einzelner Industriezweige „Message Implementation Guidelines“ (MIG) zu entwickeln, die festlegen, wie EDIFACT-Nachrichten anzuwenden sind. Doch diese MIGs sind proprietäre Richtlinien für die Mitglieder einer Interessengemeinschaft und ermöglichen keinen Datenaustausch mit Unternehmen, die andere MIGs unterstützen.

Diese Fragmentierung führt dazu, dass Netzwerkexternalitäten nur auf den unteren Ebene des EDIFACT-Standards auftreten. Jeder, der EDIFACT verwendet, kann auf der Netzwerkebene von einem anderen EDIFACT-User eine Nachricht empfangen und „parsen“, d.h. gemäß der Trennzeichen-Syntax in ihre Datenfelder zerlegen. Doch er kann nicht die Semantik, also die Bedeutung der Datenfelder, automatisch interpretieren. Auf Ebene der Semantik erzeugen nur Anwender des gleichen MIGs Netzwerkexternalitäten für ihn.

Mitschuld am relativ geringen Erfolg des EDIFACT Standards sind also seine hohe Offenheit im Sinne unseren dritten Skala und die dadurch vergebenen Netzwerkexternalitäten.

Kapitel 8

Kooperation und Kompatibilität

Dieses Kapitel beschäftigt sich mit den verschiedenen Formen von Kooperationen, die Unternehmen eingehen um die Erfolgchancen ihres Standards zu erhöhen. Nur wenige Unternehmen in Netzwerkindustrien verfügen über so eine große Marktmacht oder so ein überlegenes Produkt, dass sie ohne Hilfe von Partnern erfolgreich einen Standard durchsetzen können. Auch die größten Rivalen schließen Abkommen über gemeinsame Standards ab um das Wachstum des gemeinsamen Sektors zu beleben, so geschehen etwa zwischen Netscape und Microsoft bei der Entwicklung des SET oder OPS¹ Standards.

In diesem Kapitel geht es vorwiegend um die für die Informationstechnologie typischen Kooperationen, deren Ziel es ist einen Standard festzulegen, Kompatibilität zwischen zwei Produkten herzustellen oder allgemeiner zwei getrennte Produktnetzwerke zu einem Ganzen zu verbinden. Ausgeklammert bleiben hingegen Kooperationen, wie sie auch in der traditionellen Ökonomie auftreten, etwa um Ressourcen gemeinsam zu nutzen. Weiters muss die Kooperation nicht explizit, etwa durch einen Vertrag, abgeschlossen werden, sondern wir betrachten es auch als Kooperation, wenn zwei Unternehmen ohne explizite Absprache denselben Standard adaptieren. So sind auch offene Standards, wie sie im letzten Kapitel besprochen wurden, eine Form der Kooperation, indem sie mehreren Unternehmen ermöglichen das gleiche Netzwerk von Anwendern zu nutzen.

Zunächst werden kurz unterschiedliche Formen von Kooperationen beschrieben, um einen Eindruck über die Vielzahl der möglichen Vereinbarungen zu geben (Kapitel 8.1). Anschließend wird in Kapitel 8.2 untersucht, was Unternehmen motiviert Kooperationen einzugehen und wie Kooperationen den Wettbewerb verändern. Kapitel 8.3 stellt schließlich die Frage, welche Ausgangslagen eher zu Kooperationen und welche eher zu Inkompatibilität führen.

8.1 Formen von Kooperationen

Ganz allgemein können vertikale und horizontale Kooperationen unterschieden werden. Vertikale Kooperationen werden zwischen Anbietern verschiedener komplementärer Produkte abgeschlossen. In der Sprache unseres Schichtenmodells aus Kapitel 5 formuliert, ist das eine Kooperation zwischen Anbietern auf unterschiedlichen Ebenen. Zum Beispiel gehen CD-Produzenten mit Herstellern von CD-Playern eine vertikale Kooperation ein, indem sie beide den gleichen CD-Standard adaptieren.

Anbieter von Substituten, also von Produkten auf der gleichen Ebene, schließen dagegen horizontale Kooperationen ab. So kooperierten Sony und Philips auf horizontaler Ebene, als sie sich auf einen CD-Standard einigten, den die Player beider Unternehmen unterstützen.

Im allgemeinen haben horizontale Kooperationen den Nachteil, dass man eine Kooperation mit einem direkten Konkurrenten eingeht und diesem sein Netzwerk öffnet, während vertikale Kooperationen hauptsächlich Vorteile bringen, da dadurch mehr Komplementärgüter zur Verfügung stehen, aber die direkte Konkurrenz nicht erhöht wird. Allerdings verschwimmen oft die Grenzen zwischen horizontaler und vertikaler Kooperation, da Unternehmen auf mehreren Ebenen mit verschiedenen Produkten vertreten sind, wie wir schon am Beispiel von Microsoft gesehen haben. Für Microsoft sind andere Anbieter von Windows Anwendungen vertikale Partner im Verhältnis zum Betriebssystem, aber auch horizontale Konkurrenten im Verhältnis zu Windows-Anwendungen von Microsoft.

Egal ob Unternehmen horizontal oder vertikal verbunden sind, ist die häufigste Form der Kooperation, den gleichen Standard zu adaptieren und dadurch Kompatibilität der Produkte zu erreichen. Es gibt nun verschiedene Möglichkeiten einen solchen Standard zu entwickeln. Es

¹Der Open Profiling Standard (OPS) dient zum Schutz persönlicher Informationen im Internet.

kann etwa ein proprietärer, durch Urheberrechte geschützter Standard sein, der von einer Firma kontrolliert, aber an andere Firmen durch Lizenzen vergeben wird. Oder der Standard wurde von einer Firma entwickelt, aber wird allen anderen offen zur Verfügung gestellt. Die Kooperation kann allerdings auch schon früher einsetzen und Firmen können den Standard gemeinsam entwickeln, entweder eigenständig oder im Rahmen einer offiziellen Standardisierungsorganisation. Ebenso ist es denkbar, dass Kompatibilität einseitig erreicht wird, indem Unternehmen den Standard einer anderen Firma ohne deren Zustimmung adaptieren.

Kompatibilität kann also während der Entwicklung, der Einführung oder der Verwendung eines Standards erzeugt werden. Wenn schon mehrere Standards etabliert sind, können Firmen noch immer durch Adapter oder Konvertierungstools Kompatibilität herstellen. Meist werden Adapter von unterlegenen Firmen eingesetzt, um ihr Produkt zu einem Produkt mit größerem Netzwerk kompatibel zu machen. Oft geht die Konvertierung allerdings mit einem Qualitätsverlust einher. Von entscheidender strategischer Bedeutung ist, ob Adapter nur in eine oder in beide Richtungen funktionieren. Als Microsoft „Word 97“ einführte, erlaubte das Programm zuerst nur die Konvertierung von älteren Word-Files in das neue inkompatible Word 97-Format, aber nicht umgekehrt. Microsoft wollte damit erreichen, dass alle Anwender rasch zu Word 97 wechseln, damit sie von Word 97-Anwendern erstellte Files lesen können. Doch das Gegenteil war der Fall. Anwender wechselten nur sehr langsam zur neuen Word-Version, da sie ihre Word 97-Files nicht an andere weitergeben konnten. Als Reaktion darauf stellte Microsoft ein Plugin zur Verfügung, das das Konvertieren von Files in beide Richtungen ermöglichte[48].

Doch Standards sind nicht die einzige Möglichkeit für Unternehmen ihre Produktnetzwerke gemeinsam zu nutzen. Firmen verbinden ihre Produkte mit unterschiedlichsten Vereinbarungen. DigiMarc, ein Unternehmen, das einen Standard zur elektronischen Markierung von Bildern mit Wasserzeichen entwickelt hat, ist mit Herstellern von Bildbearbeitungsprogrammen, wie Adobe oder Corel, eine Kooperation eingegangen. Adobe und Corel verwenden in ihren Programmen eine Low-End Version der DigiMarc-Software. Im Gegensatz zu den vorigen Beispielen geht es hier zumindest für Corel und Adobe nicht darum durch einen gemeinsamen Standard höhere Netzwerkexternalitäten zu erreichen. Für sie zählt vor allem, dass sie ihre Programme kostengünstig mit einem zusätzlichen Feature ausstatten können und DigiMarc kann seine installed base dafür auf das gesamte Netzwerk von Corel und Adobe ausdehnen.

Ein anderes Beispiel ist die Kooperation zwischen eBay und Sotheby's. Beide Auktionsplattformen hatten Probleme genügend Anbieter und Käufer im Bereich von Kunstwerken und Sammlerstücke im oberen Preissegment zu finden[58] und haben daher ihre beiden Portale zu einem zusammengeführt. Das Portal ist über die Adressen ebay.com, sothebys.com und sothebys.ebay.com erreichbar. Damit profitieren beide jeweils vom bekannten Markennamen des anderen und das neue Portal startet mit den bestehenden Usern von Sotheby's und eBay als installed base. Die beiden Unternehmen schließen also ihre Netzwerke zusammen um gemeinsam die nötige kritische Masse zu erreichen.

Dass schon alleine Unterstützungserklärungen eine wertvolle Form der Kooperation sein können, hat die Java-Coalition bewiesen. Im Jahr 1997 versammelte Sun eine Reihe großer Namen der IT-Industrie, unter anderem IBM, Novell und Netscape, hinter sich, um der Durchsetzungsfähigkeit des Java-Standards Glaubwürdigkeit zu verschaffen und die Erwartungen der Anwender positiv zu beeinflussen. Sun veröffentlichte ganzseitige Zeitungsinserate mit einer Liste der Partner der Koalition[48] und veranstaltete gemeinsam mit anderen Koalitionsmitgliedern eine Promotion-Tour durch die USA[25].

Schließlich dienen auch viele Fusionen der IT-Industrie in erster Linie dem Zweck zwei Netzwerke zusammenzuführen. Microsoft hat zum Beispiel Hotmail, einen der führenden Internetmailanbieter, aufgekauft und verwendet, um für ihr „NET Passport“ Netzwerk gleich zu Beginn eine große installed base aufzubauen. Das Passport-System ermöglicht es Usern mit denselben Kundendaten verschiedene Websites zu nutzen. Jeder Hotmail-Account wurde automatisch auch zu einem Passport-Account und so verfügte das System bereits von Beginn an über eine beträchtliche Anzahl von Usern.

8.2 Wie Kooperation und Kompatibilität den Wettbewerb verändern

Der naheliegendste Grund zwei Technologien kompatibel zu gestalten ist das dadurch vergrößerte Netzwerk. Besonders wertvoll ist Kompatibilität, wenn sie unmittelbar zu einer realen Vergrößerung des Netzwerks führt und nicht nur das Potential für ein größeres Netzwerk schafft. Zu so einer realen Vergrößerung kommt es, wenn Kompatibilität zwischen Produkten mit bereits bestehenden Netzwerken geschaffen wird und diese Kompatibilität für Konsumenten auch einen Nutzen erzeugt. Wenn eine Rezeptdatenbank im Internet, wie zum Beispiel „www.thea.at“, beschließt den CookML-Standard zu unterstützen, so handelt es sich um eine reale Vergrößerung des Netzwerks. Denn Anwender können sofort auf eine größere Anzahl von Rezepten zugreifen, was für einen Rezeptstandard natürlich relevant ist.

Die Kooperation zwischen VISA und Mastercard zur Einführung des SET-Standards für sicheres Bezahlen im Internet hat hingegen nur das Potential für ein größeres Netzwerk geschaffen. Die Anzahl der Kunden, die SET prinzipiell verwenden *können*, ist dadurch zwar entscheidend höher, als wenn es zwei getrennte Standards gäbe. Doch diese Kunden müssen erst dazu motiviert werden sich für SET anzumelden und SET auch tatsächlich zu nützen, um gesteigerte Netzwerkexternalitäten zu realisieren.

Der entscheidende Vorteil eines größeren Netzwerks entsteht erst beim Auftreten von Netzwerkexternalitäten, denn dann haben Anwender durch ein gemeinsames großes Netzwerk einen höheren Nutzen. Bei starken Netzwerkexternalitäten sind Kooperationen zwischen Anbietern konkurrierender Standards oft unausweichlich, da Konsumenten ihre Kaufentscheidungen aufschieben, bis klar ist, welcher Standard gewinnt. So konnten sich 56K-Modems zum Beispiel erst durchsetzen, als sich die beiden rivalisierenden Lager auf einen gemeinsamen Standard geeinigt hatten.

In Märkten mit geringen oder keinen Netzwerkexternalitäten ist der Nutzen des gemeinsamen Netzwerks für Anbieter und Anwender hingegen nur die Summe der Nutzen der einzelnen Netzwerke.

Matutes and Regibeau[33] beschreiben aber Vorteile von Kompatibilität, die unabhängig von Netzwerkexternalitäten auftreten können. Sie modellieren das Kompatibilitätsverhalten von Unternehmen, die „Mix and Match“-Systeme anbieten. „Mix and Match“ bedeutet, dass ein System aus mehreren Komponenten zusammengesetzt wird, die über standardisierte Schnittstellen miteinander interagieren. In dem Modell können Hersteller von Systemen, die Substitute sind, entscheiden, ob sie ihre Schnittstellen kompatibel gestalten wollen oder nicht. Kompatible Schnittstellen erlauben es Anwendern Komponenten von unterschiedlichen Herstellern zu mischen, bei inkompatiblen Schnittstellen können Systeme nur aus den Komponenten eines Herstellers zusammengesetzt werden.

Ein Beispiel für so ein System ist die Hardware eines PCs. Standardisierte Schnittstellen erlauben es Komponenten verschiedener Hersteller gemeinsam zu verwenden. So können Anwender zum Beispiel eine IBM-Festplatte mit einem Asus-Mainboard und einer Matrox-Grafikkarte kombinieren und müssen nicht das ganze System von einem Hersteller beziehen.

Matutes und Regibeau zeigen, dass Kompatibilität zu höheren Verkaufspreisen der Komponenten führen können und nennen dafür zwei Ursachen. Erstens verringert Kompatibilität den Preiswettbewerb. Senkt ein Anbieter den Preis einer Komponente, so erhöht er damit auch die Verkaufszahlen der anderen Komponenten des Systems, da der Preis des Gesamtsystems dadurch sinkt. Bei Inkompatibilität müssen alle Komponenten von einem Hersteller bezogen werden, so dass er allein von der Preissenkung profitiert. Bei Kompatibilität internalisiert hingegen auch die Konkurrenz einen Teil der Preissenkung, da auch sie mehr Komponenten absetzen kann. Außerdem müssen Unternehmen ihren Standard in der Einführungsphase weniger aggressiv sponsern, wenn das Netzwerk durch Kooperationen verstärkt wird.

Der zweite Grund für höhere Preise ist, dass Kompatibilität die Nachfrage erhöht, wenn Konsumenten eine große Auswahl schätzen. Anwender können bei Kompatibilität die Systeme genauer auf ihre Bedürfnisse abstimmen, da sie zwischen Komponenten aller Hersteller auswählen können. So haben PC-Käufer einen höheren Nutzen, wenn sie sich alle Teile des PCs individuell aussuchen können, weil damit das Gesamtsystem näher an ihren Präferenzen liegt. Sonst müssten sie sich unter Umständen zwischen einem System des einen Herstellers mit einer überlegenen Grafikkarte und dem System eines anderen Herstellers mit schnellerer Festplatte entscheiden.

Der Vorteil der Kompatibilität in „Mix and Match“-Systemen ist umso größer, je mehr Anwender Produktvielfalt schätzen und je unterschiedlicher die Komponenten der verschiedenen Firmen sind. Diesem Vorteil steht aber der Nachteil höherer Konkurrenz zwischen den Firmen um die einzelnen Komponenten gegenüber. Firmen entscheiden sich für Kompatibilität, wenn die Nachfrage nach kombinierten Gütern im Verhältnis zur Nachfrage nach den eigenen Systemen relativ hoch ist[11].

In dem Modell von Matutes und Regibeau stellen alle Firmen alle Komponenten des Systems her. Werden manche Komponenten nicht von jeder Firma erzeugt oder unterscheiden sich die Komponenten so stark, dass sie keine perfekten Substitute mehr sind, erhöht sich der Anreiz für Kompatibilität, da in diesem Fall der Nachteil der größeren Konkurrenz geringer wird. Wie wir aus Kapitel 6.2 wissen, erhöht sich das Bedürfnis von Konsumenten nach Kompatibilität auch durch switching costs. Denn beim Auftreten von switching costs bevorzugen Anwender vollständigere Systeme, da sie im Falle eines Lock-Ins in den Standard mehr Komponenten zur Auswahl haben.

Auch auf Erwartungen, die Konsumenten in einen Standard oder das Netzwerk eines Standards setzen, üben Kooperationen einen positiven Einfluss aus. Denn erstens reduzieren mehrere Anbieter eines Produkts die Gefahr des Lock-Ins und zweitens erhöhen sie die Wahrscheinlichkeit, dass ein Standard sich durchsetzt. Deshalb hat Sun die schon oben erwähnte Java-Coalition gegründet und beworben. Und unter anderem auch deshalb hat im Videorecorder-Markt der von mehreren Firmen unterstützte VHS-Standard gegen Sonys Betamax-Standard gewonnen.

Doch Kooperationen und Kompatibilität bringen nicht nur diese Vorteile für Anwender und Unternehmen, sondern verändern auch den Wettbewerb grundlegend. Erstens verwandeln sie den inter-Standard- in einen intra-Standard-Wettbewerb. Unternehmen kämpfen nicht mehr darum ihren Standard im Markt durchzusetzen, sondern um einen größeren Marktanteil innerhalb des Standards.

Zweitens verlagern Kooperationen den Schwerpunkt des Wettbewerbs von Systemen zu einzelnen Komponenten. Im Markt der Videospiele, in dem zum Beispiel die Systeme von Nintendo

und Sega inkompatibel sind, entscheiden sich Konsumenten zwischen den beiden Firmen, je nachdem welches komplette System besser ist. Sie beziehen also sowohl verfügbare Spiele als auch Qualität der Hardware in ihre Entscheidung mit ein. Beim Kauf eines CD-Players muss sich hingegen niemand überlegen, mit welchem Modell er die bessere Musik hören kann, sondern wählt ihn allein auf Grund der überlegenen Hardwareeigenschaften aus.

Drittens setzen Unternehmen in Märkten mit Kompatibilität eher Methoden des Preiswettbewerbs als der Produktdifferenzierung ein. Denn der Spielraum für eine Differenzierung von Konkurrenzprodukten ist durch die Bindung an einen gemeinsamen Standard eingeschränkt. Dieser Tendenz wirkt allerdings entgegen, dass bei Kompatibilität, wie oben beschrieben, Preissenkungen auch von anderen Unternehmen internalisiert werden. Der hier beschriebene Effekt des erhöhten Preiswettbewerbs spielt vor allem bei der horizontalen Konkurrenz zwischen Anbietern von funktional ähnlichen, substituierbaren Gütern eine Rolle. Der oben diskutierte Effekt des sinkenden Preiswettbewerbs ist vor allem relevant, wenn Kompatibilität zwischen Anbietern von Komplementärgütern vereinbart wird. Denn vertikal verbundene Unternehmen konkurrieren nicht direkt miteinander, profitieren aber von der Preissenkung der Konkurrenten über das billigere Gesamtsystem.

Dass Unternehmen ihre Produkte innerhalb einer Kooperation weniger differenzieren können, ist einer der Gründe Inkompatibilität vorzuziehen. Oft versuchen sich aber auch Firmen durch das Erstellen von proprietären Erweiterungen zum Standard von Konkurrenten zu unterscheiden. Solche Veränderungen führen aber, wie zum Beispiel bei den Erweiterungen des HTML-Standards durch Microsoft und Netscape, oft zu einer Fragmentierung des Standards.

Die Notwendigkeit sich an vereinbarte Kompatibilitätsbedingungen zu halten kann weiters Produktvielfalt reduzieren und Innovationen hemmen. Denn vor allem wenn Unternehmen den Standard nicht alleine kontrollieren, können neue Produktfeatures nur soweit implementiert werden, wie dies die gemeinsam vereinbarte Spezifikation des Standards zulässt.

Die in diesem Kapitel beschriebenen Effekte von Kooperationen werden in der folgenden Liste von Kompatibilitäts-Faktoren noch einmal zusammengefasst. Jeder der Faktoren beeinflusst den Grad der Kompatibilität, der für eine Technologie optimal ist:

1. Wie stark sind die Netzwerkeexternalitäten und wie wichtig ist es für Anwender, dass es nur *einen* Standard gibt?
2. Wie stark ist der Lock-In für Kunden und wie sehr sind sie auf diese Gefahr sensibilisiert?
3. Wie viel Nutzen erzeugt „Mix and Match“ von Komponenten unterschiedlicher Hersteller?
4. Wie groß wären die Vorteile eines proprietären Standards, wie zum Beispiel alleinige Kontrolle über den Standard oder mögliche Lizenznahmen?
5. Wie stark ist die Position des eigenen Standards im Verhältnis zu den Konkurrenten in den Bereichen
 - Zeitvorsprung
 - installed-base
 - Marktmacht (Anzahl von erreichbaren Kunden, Bekanntheitsgrad, Image usw.)
6. Wer profitiert mehr von den durch eine Kooperation veränderten Rahmenbedingungen, d.h.:

- Ist das eigene Unternehmen für einen Wettbewerb nach Preis- oder nach Produktdifferenzierung besser gerüstet? Wie viele Innovationsmöglichkeiten werden durch Kompatibilität vergeben?
- Ist die Konkurrenz auf System- oder auf Komponentenebene erfolversprechender? Zum Beispiel wäre Inkompatibilität zu bevorzugen, wenn eine Komponente des Systems der Konkurrenz so weit überlegen ist, dass alle Anwender nur wegen der einen Komponente das ganze System kaufen.

8.3 Die Kooperationsentscheidungen mehrerer Unternehmen

Die im letzten Kapitel definierten Faktoren bestimmen, ob ein Unternehmen eine Kooperation mit anderen Unternehmen eingehen will. Damit die Koalition zustande kommt, müssen aber auch die potentiellen Partner daran Interesse haben. In diesem Kapitel werden unterschiedliche Interessenskombinationen anhand eines von Besen und Farrell [4] entwickelten Modells analysiert.

Grundlage des Modells für zwei Unternehmen ist die in Tabelle 8.1 dargestellte Payoff-Matrix. Sie gibt die Payoffs² für die Unternehmen A und B abhängig davon an, welche Technologie jedes Unternehmen adaptiert. Unternehmen A erhält zum Beispiel den Nutzen a_{12} , wenn A Technologie 1 und B die Technologie 2 adaptiert. Wir nehmen an, dass Technologie 1 vom Unternehmen A und Technologie 2 von Unternehmen B entwickelt wurde. Die linke untere Ecke beschreibt also die Payoffs für den Fall, dass jedes Unternehmen die Technologie des jeweils anderen adaptiert. Unter der realistischen Annahme, dass es entweder vorteilhaft ist seine eigene Technologie zu verwenden oder eine Kooperation einzugehen, wird dieser Fall immer von einer anderen Lösung dominiert und ist daher vernachlässigbar. Der Fall 11 und der Fall 22 können als Kooperation oder zumindest als Kompatibilität zwischen den beiden Technologien interpretiert werden, während im Fall 12 zwei inkompatible, konkurrierende Technologien im Markt eingeführt werden.

		Unternehmen B	
		Technologie 1	Technologie 2
Unternehmen A	Technologie 1	a_{11}, b_{11}	a_{12}, b_{12}
	Technologie 2	(a_{21}, b_{21})	a_{22}, b_{22}

Tabelle 8.1: Payoff-Matrix für die Kooperationsentscheidung zweier Unternehmen

Die Höhe der Payoffs der unterschiedlichen Fälle wird durch die im vorigen Kapitel herausgearbeiteten Faktoren bestimmt. Analysieren wir die Situation aus Sicht der Firma A, so beeinflussen die Faktoren 1 bis 4 die Payoffs a_{11} und a_{22} relativ zu a_{12} . Je stärker die Netzwerkeexternalitäten, je größer die Angst der Anwender vor Lock-In und je höher der Nutzen aus „Mix and Match“, umso höher ist der Nutzen von Kompatibilität im Vergleich zu Inkompatibilität. Die ersten drei Faktoren beeinflussen die Payoffs beider Unternehmen im gleichen Maße, weil sie Eigenschaften des gesamten Marktes wiedergeben. Der Faktor 4 verändert die Payoffs der Unternehmen unterschiedlich, da sich Vorteile eines proprietären Standards, wie zum Beispiel Lizenzeinnahmen, zwischen den beiden Unternehmen unterscheiden können.

²Nutzenwerte

Die eigene Marktstellung (Faktor 5) beeinflusst den Payoff a_{12} im Verhältnis zu a_{22} und im Verhältnis zu b_{12} . Denn wenn das Unternehmen A eine stärkere Stellung im Markt hat als das Unternehmen B, dann kann es bei Inkompatibilität eher seinen eigenen Standard am Markt durchsetzen. Dadurch sinkt einerseits die Attraktivität nachzugeben und die Technologie des anderen zu adaptieren (a_{22}) und andererseits der Payoff von Inkompatibilität für den schwächeren Konkurrenten (b_{12}).

Faktor 6 beschreibt Vor- bzw. Nachteile eines Unternehmens im Wettbewerb innerhalb einer Kooperation. Er verändert dementsprechend a_{11} relativ zu b_{11} und a_{22} relativ zu b_{22} . Denn ist Unternehmen A für einen Preiswettbewerb besser gerüstet als für Produktdifferenzierung oder kann am Komponentenmarkt besser reüssieren als am Systemmarkt, so kann es von der Kompatibilität mehr profitieren als Unternehmen B.

Je nach Verteilung der Payoffs kann das Modell vier unterschiedliche Ergebnisse haben:

1. Beide Unternehmen bevorzugen den gleichen Standard. (z.B. $a_{22} > a_{11}, a_{12}$ und $b_{22} > b_{11}, b_{12}$)
2. Beide Unternehmen wollen ihren eigenen Standard durchsetzen und ziehen Inkompatibilität einer Kooperation vor. ($a_{12} > a_{22}$ und $b_{12} > b_{11}$)
3. Beide Unternehmen möchten ihren eigenen Standard adaptieren, präferieren aber eine gemeinsame Adaption des fremden Standards gegenüber Inkompatibilität. ($a_{11} > a_{22} > a_{12}$ und $b_{22} > b_{11} > b_{12}$)
4. Ein Unternehmen bevorzugt Inkompatibilität und das andere Kompatibilität. (z.B. $a_{11}, a_{22} > a_{12}$, aber $b_{12} > b_{11}, b_{22}$)

Der erste Fall kann eintreten, wenn zumindest eines der Unternehmen noch keine großen Investitionen in einen eigenen Standard getätigt hat oder in einen neuen Marktbereich einsteigt und daher keine Probleme hat sich auf einen gemeinsamen Standard zu einigen. Ebenso kann es sein, dass ein Standard dem anderen klar überlegen ist und der unterlegene Anbieter deshalb bereitwillig den besseren Standard adaptiert.

Im zweiten Fall wollen beide Unternehmen ihren eigenen Standard durchsetzen. Da keiner bereit ist den fremden Standard zu adaptieren, werden beide Technologien eingeführt und es kommt zu einem Kampf um die Vorherrschaft im Markt. Mögliche Gründe für so einen Ausgang sind geringe Netzwerkexternalitäten oder sehr hohe Gewinnerwartungen aus einem proprietären Standard.

Die dritte Variante entsteht, wenn es sehr wichtig ist Kompatibilität zu erreichen, zum Beispiel weil Netzwerkexternalitäten sehr stark sind und Anwender nur *einen* Standard akzeptieren würden. Diese Situation entspricht einer „battle of the sexes“³. Jedes Unternehmen versucht das andere in einer Verhandlungsphase zu überzeugen den eigenen Standard zu adaptieren. Während der Verhandlung können Bekenntnisse zur eigenen Technologie (commitments) und Zugeständnisse (concessions) strategisch eingesetzt werden.

Commitments erhöhen den eigenen Payoff für die Wahl des eigenen Standards (a_{11}) gegenüber dem eigenen Payoff für die Wahl des Konkurrenzstandards (a_{22}). Zum Beispiel könnte ein Unternehmen während der Verhandlungsphase in seine installed base oder in die Weiterentwicklung des eigenen Standards investieren. Ziel ist es dem Konkurrenten zu zeigen, dass man

³Die „battle of the sexes“ ist ein klassisches Problem der Spieltheorie, bei dem ein Ehepaar versucht sich auf eine Abendgestaltung zu einigen. Er würde lieber zu einem Fußballmatch, sie aber lieber in die Oper gehen. Beide ziehen es aber vor den Abend gemeinsam, statt allein bei ihrer Lieblingsbeschäftigung zu verbringen. Es kommt zu einer Verhandlung, bei der jeder versucht den anderen umzustimmen.

entschlossen ist den eigenen Standard einzuführen. Wenn man dem Konkurrenten glaubhaft machen kann, dass man sich schon für die eigene Technologie entschieden hat, so bleibt diesem nichts anderes übrig als diesen Standard ebenfalls zu adaptieren. Denn würde er sich für seinen Standard und damit für Inkompatibilität entscheiden, hätte er einen geringeren Payoff.

Zugeständnisse erhöhen den Payoff des anderen Unternehmens, wenn es einer Adaption des fremden Standards zustimmt (b_{11}). Dadurch wird es der Konkurrenz leichter gemacht von ihrem eigenen Standard abzurücken. Solche Zugeständnisse können zum Beispiel niedrige Lizenzgebühren, ein offener Entwicklungsprozess oder eine Kombination der beiden Standards sein. All diese Zugeständnisse senken jedoch wiederum den eigenen Payoff, wenn schließlich der eigene Standard gewinnt (a_{11}).

Im vierten Fall möchte zum Beispiel Unternehmen A Kompatibilität herstellen, doch B bevorzugt Inkompatibilität. Das heißt, B will verhindern, dass Unternehmen A seinen Standard adaptiert. Dieser Fall tritt vor allem dann auf, wenn Unternehmen B schon eine installed base aufgebaut hat oder so mächtig ist, dass es keine Kooperationspartner benötigt. Eine Kooperation würde für Unternehmen B nur den Wettbewerb verstärken und Einnahmen kosten, aber keinen wirklichen Vorteil bringen. Unternehmen A ist meist ein unterlegenes Unternehmen, das zu spät am Markt eingetroffen ist oder aufgrund seiner schwachen Marktposition nicht ohne Partner überleben kann. Das Ergebnis dieser Situation hängt davon ab, ob Unternehmen B über Mittel verfügt um das Konkurrenzunternehmen von seinem Netzwerk auszuschließen. Verschiedene Beispiele für solche Mittel, wie Urheberrechte oder geheime Interface-Spezifikationen, wurden in dieser Arbeit schon ausführlich besprochen.

Die Ausführungen dieses Modells sind natürlich vereinfacht. So erlauben sie nur die Analyse von zwei Unternehmen und bilden auch nicht die Unsicherheit bei der Schätzung der Payoffs des eigenen und des anderen Unternehmens ab. Doch erstens lässt sich das Modell relativ einfach erweitern und zweitens ist es ohnehin nur schwer möglich die Effekte einer Kooperation genau zu beziffern und daraus quantitativ korrekte Payoffs zu berechnen. Das Modell dient viel mehr dazu das Zusammenspiel zwischen den relevanten Faktoren aufzuzeigen und Unternehmen ein Framework für ihre Kooperationsüberlegungen zu geben.

De Vries und Hendrikse [10] analysieren das Kooperationsverhalten der niederländischen Banken bei der Einführung eines Chipkartenstandards für elektronisches Bezahlen. Sie verwenden dazu ein spieltheoretisches Modell, das auch Unsicherheit über die Akzeptanz von Chipkarten durch die Konsumenten miteinbezieht. Allerdings berücksichtigt das Modell nur die Kompatibilitätsentscheidung eines Unternehmens und behandelt nicht den Fall der Inkompatibilität. Wir werden nun die Daten aus [10] verwenden, um die Überlegungen beider Bankenkonsortien anhand unseres Frameworks zu untersuchen.

Im Jahr 1994 beschlossen alle niederländischen Banken einen gemeinsamen Standard für die Bezahlung mit Chipkarten zu entwickeln. Ihre Hauptmotive waren Kostenersparnis und eine Vereinfachung der EURO-Einführung. Der gemeinsame Standard mit dem Namen Chipknip wurde 1995 in einem Pilotprojekt in Arnheim eingeführt. Doch im Dezember 1995 entschied die ebenfalls am Chipknip-Standard beteiligte Postbank gemeinsam mit dem größten niederländischen Telekommunikationsunternehmen KPN einen eigenen Chipkarten-Standard namens Chipper einzuführen. In den darauf folgenden Jahren kämpften die beiden Standards um die Vorherrschaft im Markt und der Chipknip-Standard konnte einen leichten Vorsprung herausarbeiten. Jedoch konnte sich letztendlich keiner der beiden Standards durchsetzen. Der Kampf zwischen den Systemen sorgte für Unsicherheit bei den Konsumenten und die Nutzungsraten der Karten blieben zum Beispiel im Vergleich zu Belgien äußerst gering. 1998 begann eine Annäherung zwischen den beiden Systemen, zunächst wurde eine gemeinsame Benutzung der Ladeterminale vereinbart

und 1999 wurden auch Terminals für die Bezahlung kompatibel gestaltet. Im März 2001 wurde schließlich vereinbart alle Chipper-Karten durch die Chipknip-Technologie zu ersetzen und über eine gemeinsame technologische Infrastruktur zu verwalten. Die Ausgangssituation von 1995 war wieder erreicht.

Der Kampf zwischen den beiden Standards hatte die Verbreitung der Chipkartenbezahlung verzögert, ohne einem der Unternehmen oder den Anwendern irgendeinen Vorteil zu bringen. Die Entwicklung zweier Systeme und die anschließende Zusammenführung durch Gateway-Lösungen hatte hohe, unnötige Kosten aufgeworfen.

Doch wie ist diese Situation entstanden? In unserer Analyse sollen die Chipper-Kooperation von Postbank und KPN Gruppe A und die Chipknip Banken Gruppe B sein. Der Fall 22 steht für die ursprünglich vereinbarte gemeinsame Kooperation und im Fall 12 unterstützt jedes Unternehmen seinen eigenen Standard. Die Chipper-Gruppe hat den anderen Banken von Anfang an angeboten den Chipper-Standard über Lizenzierung zu nutzen, das wäre Fall 11.

Postbank und KPN konnten zu Beginn drei Vorteile für sich verbuchen. Erstens war die Chipper-Technologie Chipknip überlegen und erlaubte die Nutzung von mehr Zusatzfunktionen, wie zum Beispiel Treueprogrammen oder Speicherung von Identifikationsmerkmalen. Zweitens wurde die Telekommunikationsinfrastruktur von KPN verwendet, um Chipkarten einerseits in Telefonzellen und andererseits mit Heimtelefonen aufladen zu können. Drittens hatten sowohl Postbank als auch KPN eine große installed base von Privatkunden. Im Unterschied zu Chipknip würde Chipper auch in Telefonzellen als Telefonwertkarte verwendbar sein. Acht Millionen Personen hatten bereits eine Telefonwertkarte und würden diese Karten zumindest teilweise durch Chipper-Karten ersetzen. Jeder dieser drei Vorteile hatte zwei Effekte. Erstens steigerte er die Chance von Chipper sich im Kampf gegen Chipknip durchzusetzen. Er erhöhte also den Payoff a_{12} gegenüber b_{12} . Zweitens steigerte er aber auch die Attraktivität des Chipper-Standards für andere Banken. Er erhöhte also b_{11} gegenüber b_{22} .

Dem standen aber auch zwei Vorteile des Chipknip-Netzwerkes gegenüber. Dieses hatte einen first-mover advantage von einem dreiviertel Jahr. Dieser Vorteil senkte a_{12} gegenüber a_{22} , erhöhte aber b_{12} gegenüber b_{11} . Außerdem hat der Großteil der niederländischen Geschäfte Bankverbindungen mit einer Bank der Chipknip-Gruppe und adaptierte daher eher das Chipknip-System. Dadurch wurde die Stellung von Chipknip in einem Kampf um die Vorherrschaft gestärkt und damit a_{12} gesenkt beziehungsweise b_{12} erhöht. Weiters steigt die Attraktivität einer Kooperation für die Chipper-Unternehmen, also sowohl a_{11} als auch a_{22} , da sie durch eine Kooperation Geschäftskunden besser erreichen können.

Diese Attraktivität einer Kooperation lässt sich auch so erklären, dass die Postbank für einen Wettbewerb auf Komponentenebene besser gerüstet ist als für einen Wettbewerb um ganze Systeme. Ein System besteht aus einer Karte, einem dazu kompatiblen Ladegerät und einem kompatiblen Abbuchungsterminal in einem Geschäft. Die Postbank hat zwar viele Privatkunden, aber nur eine geringe Anzahl an Geschäftskunden. Dadurch verliert das gesamte System an Wert, denn ein Bezahlungssystem ohne Geschäfte, die diese Bezahlung akzeptieren, ist wertlos. So geht auch der Vorteil verloren, den die Postbank bei einer der Komponenten, der Anzahl der verteilten Chipper-Karten, hat.

Nach diesen Überlegungen entschied sich die Postbank dafür Chipper einzuführen und die Chipknip-Kooperation zu verlassen. Da die Chipper-Gruppe von Anfang an bereit war die anderen Banken durch Lizenzen an Chipper zu beteiligen, wissen wir, dass sie eine allgemeine Unterstützung des Chipper-Standards gegenüber Inkompatibilität mit den anderen Banken bevorzugte ($a_{11} > a_{12}$). Weiters muss a_{11} größer als a_{22} gewesen sein, denn sonst hätte sie keinen

eigenen Standard eingeführt. Dass sie einen eigenen Standard bevorzugte, kann an den Lizenzeinnahmen, mehr Kontrolle über den Standard oder der Kooperation mit KPN liegen, die vielleicht mit den anderen Banken nicht zustande gekommen wäre.

Die wichtige Frage ist nun, ob a_{12} zum Zeitpunkt der Abspaltung größer war als a_{22} ; ob die Chipper-Gruppe also Inkompatibilität einer gemeinsamen Chipknip-Implementierung vorzog. Ex post wissen wir, dass a_{12} kleiner war als a_{22} . Denn nach Jahren der Inkompatibilität schloss sich die Chipper-Gruppe ja wieder dem Chipknip-Netzwerk an, entschied sich also für den Fall 22.

War sich die Postbank von Anfang an im Klaren darüber, dass a_{12} kleiner ist als a_{22} , so hat sie mit ihrer Konkurrenz eine kostspielige „battle of the sexes“ ausgetragen. Es war ihr Ziel die Konkurrenz zu überzeugen zum eigenen Chipper-Standard zu wechseln und so den höheren Payoff a_{11} zu realisieren. Für die andere Seite war vermutlich umgekehrt b_{12} kleiner als b_{11} , auch sie hätten jede Form der Kompatibilität der Inkompatibilität vorgezogen. Doch die Gruppen benötigten mehrere Jahre bis sie sich auf ein System einigen konnten.

Allerdings kann es auch sein, dass die Postbank die Situation ex ante falsch eingeschätzt hat und annahm, dass sogar Inkompatibilität besser war als Chipknip zu akzeptieren ($a_{12} > a_{22}$). Inkompatibilität ist vor allem auch deswegen eine schlechte Lösung, weil alle Kunden bereits auf eines der beiden Systeme festgelegt waren. Denn fast jede Person und jedes Unternehmen hat bereits ein Bankkonto bei einer Chipknip oder einer Chipper-Bank. Keiner der beiden Chipkarten-Standards war dem anderen soweit überlegen, dass sein Mehrwert die switching costs eines Bankwechsels ausgleichen konnte. Das heißt, so lange Inkompatibilität vorherrscht, kann keines der beiden Systeme alle User gewinnen und es wird immer zwei getrennte Systeme geben. Die Netzwerkexternalitäten für Chipkarten sind allerdings sehr stark, denn für Anwender ist es inakzeptabel ihre Karte nur in der Hälfte der Geschäfte einsetzen zu können. Weitere Probleme entstehen dort, wo Chipkarten die einzige Bezahlungsform darstellen, wie zum Beispiel bei manchen Parkautomaten. Daher hat ein Chipkartenmarkt mit zwei konkurrierenden Systemen nur wenig Erfolgchancen, wie ja auch die Praxis gezeigt hat. Die Payoff-Werte von a_{12} und b_{12} liegen damit nahe Null. Eine Fehleinschätzung von a_{12} durch die Postbank kann zwei Gründe haben. Entweder sie haben die Netzwerkexternalitäten unterschätzt und angenommen es wäre auch Platz für zwei Standards. Oder sie haben den Nutzen der Chipkarte für Anwender und Unternehmen höher eingeschätzt, sodass auch Kunden der Chipknip-Gruppe beginnen würden Chipper zu nutzen.

Egal ob „battle of the sexes“ oder bewusste Inkompatibilität haben sich im Nachhinein auch die Wettbewerbsvorteile der Postbank als wesentlich geringer herausgestellt. Der Vorteil der größeren Anzahl von Ladestationen dank des KPN-Netzwerks wurde dadurch reduziert, dass die Chipknip-Gruppe bald ebenfalls Ladestationen für daheim anbot. Außerdem waren Kunden eher daran gewöhnt Geld vom Bankomaten zu holen, wo sie auch ihre Chipknip-Karten aufladen konnten als aus Telefonzellen. Schließlich verlor die KPN auch noch ihr staatliches Monopol und neue Telekommunikationsunternehmen schlossen sich Chipknip an. Weiters wurden die Chipknip-Karten ebenfalls für neue Funktionen erweitert und so verringerte sich auch die technische Überlegenheit von Chipper.

Zusammenfassend kann man feststellen, dass in diesem Fall die Netzwerkfaktoren über überlegene Produktfeatures gesiegt haben. Produktvorteile konnten von der Konkurrenz imitiert werden. Doch die Netzwerkexternalitäten bestrafte Inkompatibilität, die switching costs beim Wechsel des Netzwerks waren zu stark, die Gruppe mit dem first-mover advantage konnte das größere Netzwerk aufbauen und letztendlich war Kompatibilität die einzige effiziente Lösung.

Teil III

Entscheidungsframework und Ausarbeitung einer Strategie

Kapitel 9

Synthese der Grundlagen zu einem Entscheidungsframework

In diesem dritten Teil sollen die im Teil II einzeln präsentierten Aspekte von Standards in Netzwerkindustrien zusammengeführt und am Beispiel des Rezeptstandards angewendet werden.

Zunächst wird ein Framework entwickelt, das den Zusammenhang zwischen den verschiedenen Faktoren von Einführungsstrategien darstellt und Unternehmen eine Hilfe zur systematischen Entwicklung ihrer Einführungsstrategie bietet.

Wie wir gesehen haben, besteht eine Einführungsstrategie für einen Standard aus zahlreichen Elementen, die aufeinander und auf vorgegebene Umweltbedingungen abgestimmt werden müssen. Das Framework zeigt auf, welche Elemente existieren, in welcher Reihenfolge sie behandelt werden sollen und wie sie sich gegenseitig beeinflussen. Der Anwender des Frameworks muss schließlich entscheiden, wie wichtig diese Elemente und die Beziehungen zwischen den Elementen im konkreten Fall sind.

In Kapitel 9.1 wird der Aufbau des Frameworks erklärt und in den drei darauf folgenden Kapiteln werden die einzelnen Teile des Frameworks detailliert beschrieben. Im Kapitel 10 wird die Anwendung des Frameworks Schritt für Schritt an der Entwicklung einer Einführungsstrategie für den Rezeptstandard demonstriert.

9.1 Beschreibung des Frameworks

Die Entwicklung einer Einführungsstrategie für einen Standard teilt sich nach unserem Framework grob in drei Phasen (siehe Abbildung 9.1).

Zuerst wird die Marktsituation zum Einführungszeitpunkt untersucht. Dabei wird die Stärke der Netzwerkexternalitäten, die Gefahr von excess inertia und die Bedrohung durch mögliche Konkurrenz bewertet. Ebenso werden standard-spezifische Faktoren, wie switching costs oder die Möglichkeit Konkurrenten auszuschließen bestimmt.

In der zweiten Phase werden aufbauend auf diesem Wissen die Grundpfeiler der Strategie definiert. Es wird festgelegt, welches Ziel mit dem Standard erreicht werden soll, wie offen der Standard gestaltet wird und welche Form von Kooperation und Kompatibilität gewählt werden soll.

Die dritte Phase dient nun der Feinabstimmung der Strategie. Dabei werden sowohl in der Marktanalyse festgestellte Probleme weiter behandelt als auch die Elemente der groben Strategie verfeinert. So werden zum Beispiel zusätzliche Maßnahmen gegen excess inertia oder gegen die Gefahr einer Fragmentierung bei einem offenen Standard entwickelt.

Die Phasen sind jeweils durch eine Art Schnittstelle verbunden, die angibt, welche Faktoren aus der vorigen Phase Entscheidungen der nächsten Phase beeinflussen.

Das Framework enthält nur jene Einflussfaktoren und strategische Entscheidungen, die aus Sicht des Standards oder der netzwerkökonomischen Überlegungen relevant sind. Dazu kommen bei jeder Produkteinführung natürlich noch zahlreiche andere betriebswirtschaftliche Überlegungen, etwa des Marketings oder der Produktion. Diese Überlegungen schränken gemeinsam mit den Eigenschaften des einzuführenden Produkts den Entscheidungsraum weiter ein. So war es zum Beispiel für die Entwickler von PayPal trivialerweise keine Option den Standard nur einzuführen um die Nachfrage im Markt zu steigern, da sie selbst noch nicht im Markt vertreten waren. Damit standen ihnen von vornherein nicht alle Möglichkeiten der Finanzierungsentscheidung offen.

Die strategischen Entscheidungen wurden in zwei Phasen aufgeteilt, um die Komplexität bei jeder Entscheidung zu reduzieren. Die Entscheidung für eine grobe Strategie in der zweiten Phase ist weitgehend unabhängig von der Feinabstimmung der dritten Phase. Trotzdem kann in der Praxis eine inkrementelle Vorgehensweise sinnvoll sein. Nach der dritten Phase könnten

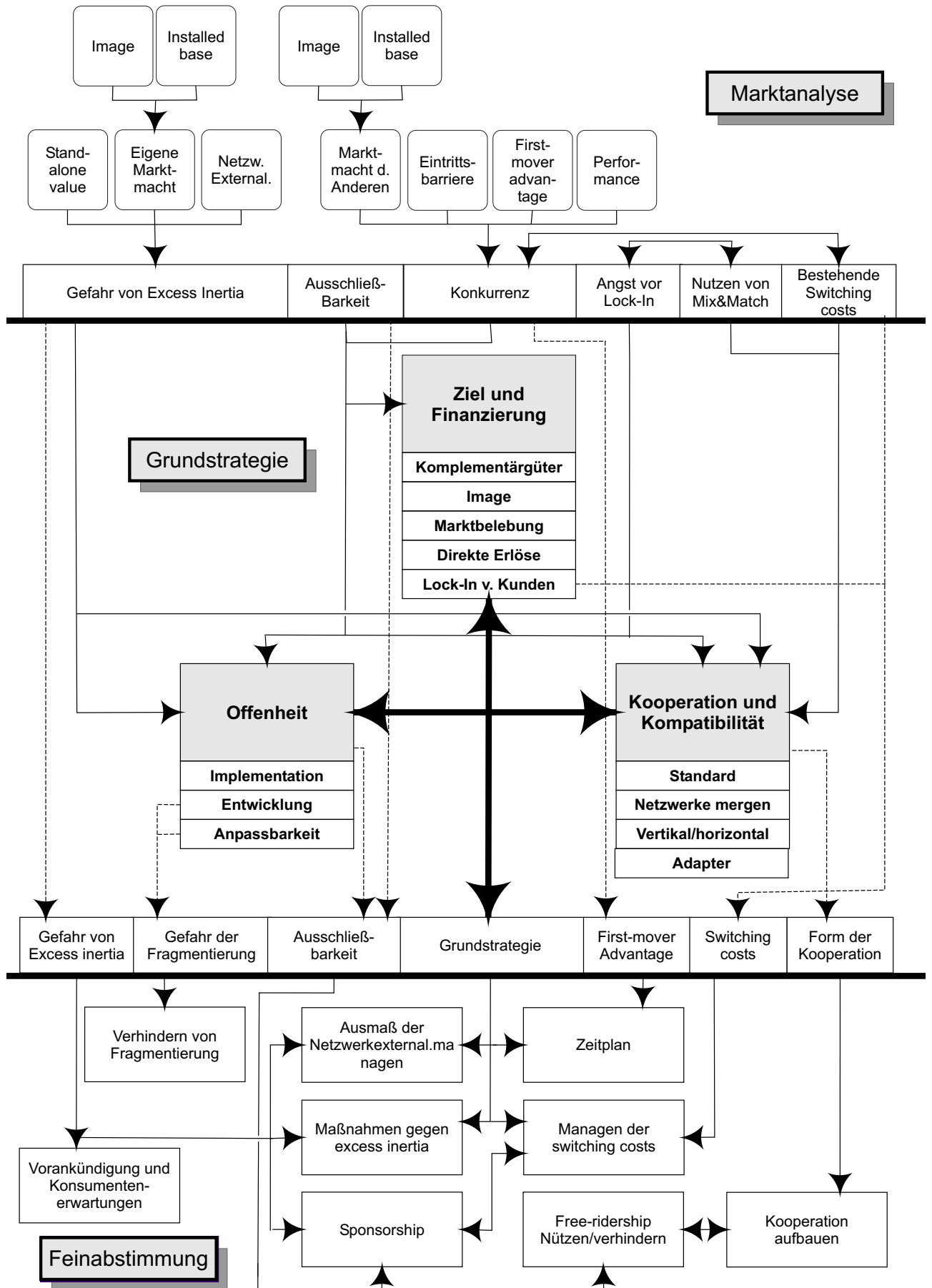


Abbildung 9.1: Graphische Darstellung des Entscheidungsframeworks

zum Beispiel noch Änderungswünsche an der Offenheitsstrategie auftreten, weil die Gefahr der Fragmentierung zu groß erscheint oder zu wenig Mittel gegen excess inertia gefunden wurden.

9.2 Phase 1 - Marktanalyse

In dieser Phase werden die verschiedenen den Standard beeinflussenden Faktoren untersucht und beurteilt, ob sie in der konkreten Situation relevant sind. Die Faktoren beeinflussen die Strategieentscheidungen in unterschiedlichem Ausmaß. Manche betreffen nur einzelne Entscheidungselemente, andere liegen allen Entscheidungen zu Grunde. Die Einflussfaktoren sind noch unabhängig von den Strategieentscheidungen der nächsten beiden Phasen, sie geben lediglich Eigenschaften des Marktes wieder.

Bei den komplexeren Einflussfaktoren „excess inertia“ und „Konkurrenz“ werden in Abbildung 9.1 elementare Eigenschaften angegeben, die diese Einflussfaktoren wesentlich bestimmen. Die folgenden Einflussfaktoren werden im Framework verwendet:

- **Excess inertia:** Excess inertia beschreibt den Effekt, dass ein Standard sich nicht durchsetzen kann, weil sein Netzwerk zu Beginn zu klein ist und er daher Anwendern zu wenig Nutzen bietet (siehe Kapitel 4). Die Gefahr von excess inertia ist umso größer, je größer die Bedeutung der Netzwerkexternalitäten und je kleiner der stand-alone value des Standards ist. Es ist wichtig alle Quellen von Netzwerkexternalitäten zu erkennen, direkte und indirekte, und zu beurteilen, durch welche dieser Quellen excess inertia ausgelöst wird. So können zielgenauere Maßnahmen entwickelt werden. Bekanntheitsgrad eines Unternehmens und eine vorhandene installed base können die Wahrscheinlichkeit von excess inertia reduzieren, diese Faktoren wurden im Begriff „Marktmacht“ zusammengefasst. Müssen Anwender switching costs überwinden um zum neuen Standard zu wechseln, erhöht sich hingegen die Gefahr von excess inertia, da der stand-alone value weiter gesenkt wird.
- **Konkurrenz:** Ein Standard muss sich möglicherweise nicht nur gegen excess inertia sondern auch gegen einen zweiten konkurrierenden Standard durchsetzen. Dieser Faktor beschreibt die Konkurrenz durch bereits bestehende Standards, durch Standards die gleichzeitig eingeführt werden, aber auch das Potential für zukünftige Konkurrenzstandards. Wie gefährlich die Konkurrenz ist, hängt davon ab, ob switching costs beim Wechsel auftreten, wer über das größere Netzwerk verfügt, ob ein Standard einen first-mover advantage hat, hinter welchem Standard eine größere Marktmacht steht und wessen Produkt überlegen ist. Die Eintrittsbarriere in den Markt gibt an, wie hoch die Wahrscheinlichkeit einer zukünftigen Konkurrenz ist. Im Programmiersprachen-Markt besteht durch den hohen Entwicklungsaufwand etwa eine relativ hohe Barriere für neue Konkurrenten. Dagegen lässt sich ein neues XML-Format für Rezepte vergleichsweise schnell entwickeln.
- **Ausschließbarkeit:** Für die Wahl einer Strategie ist es von grundlegender Bedeutung zu wissen, ob und mit welchen Mitteln Konkurrenten vom Standard ausgeschlossen werden können (siehe Kapitel 5.2). Ausschließbarkeit ermöglicht erst die Einhebung von Lizenzgebühren und kann Unternehmen weitere Wettbewerbsvorteile gegenüber Konkurrenten verschaffen. Hier kann zum Beispiel analysiert werden, welche rechtlichen Maßnahmen, wie etwa Patentschutz, einem Unternehmen zur Verfügung stehen.
- **Angst vor Lock-In:** Die Einführung eines Standards kann auch dadurch gehemmt werden, dass Anwender Lock-In in den Standard befürchten und deshalb den Standard nur zögerlich adaptieren. Anwender wollen vor allem Lock-In in einen Standard vermeiden, bei

dem nicht sicher ist ob er jemals eine ausreichende Netzwerkgröße erreichen, beziehungsweise sich gegen einen Konkurrenzstandard durchsetzen wird. Deshalb wiegt dieser Punkt bei kleinen, unbekanntem Unternehmen, in die Anwender noch wenig Vertrauen haben, besonders schwer.

- **Nutzen von Mix and Match:** Dieser Faktor beschreibt den Nutzen, den Anwender dadurch erhalten, dass sie Komponenten verschiedener Hersteller miteinander kombinieren können. Wie im Kapitel 8.2 erklärt, können Anwender Produkte durch Kompatibilität genauer auf ihre Bedürfnisse abstimmen und erhalten dadurch einen höheren Nutzen. Mix und Match Systeme lindern auch die Angst der Konsumenten vor Lock-In, da sie ohne switching costs Produkte mehrerer Anbieter beziehen können.

Diese Faktoren ergeben gemeinsam die Umgebung, in der die Entscheidungen der nächsten beiden Phasen getroffen werden.

9.3 Phase 2 - Das Grundgerüst der Strategie

In der zweiten Phase wird das Grundgerüst der Strategie festgelegt. Dazu muss zu allererst geklärt werden, welche Möglichkeiten es für das einführende Unternehmen gibt den Standard zu finanzieren. Dann werden anhand der ausgewählten Ziele des Standards eine detaillierte Offenheits- und Kooperationsstrategie ausgearbeitet. Nebenbedingungen für diese Entscheidungen sind die in Phase 1 definierten Einflussfaktoren.

Unternehmen verfolgen mit der Einführung eines Standards verschiedenste **Ziele** und haben unterschiedliche Strategien, wie sie von dem Standard profitieren können. Die hier aufgezählten sind nur häufige Beispiele oder Elemente einer Gesamtstrategie und wurden alle in dieser Arbeit besprochen. Deshalb werden sie hier nur mehr kurz beschrieben:

- Von Komplementäreffekten profitieren Unternehmen, wenn sie Produkte auf einer Ebene verbilligt oder gratis anbieten, aber dadurch auf der dazu komplementären Ebene mehr Produkte verkaufen können. So finanziert zum Beispiel Adobe den kostenlosen Acrobat PDF-Reader mit dem Verkauf von Programmen zum Erstellen von PDF-Dateien (siehe Kapitel 5.4).
- Manchmal genügt schon eine uneigennützig Belegung des gesamten Marktes als Ziel für eine Standardeinführung. Ein Standard wird eingeführt um den Kunden einen höheren Nutzen bieten zu können und so die Nachfrage zu erhöhen. Doch es wird nicht nur die Nachfrage nach eigenen Produkten sondern auch die Nachfrage nach Produkten von Konkurrenzunternehmen erhöht (siehe Kapitel 5.4). So belebten Sony und Philips mit der Einführung des CD-Standards den stagnierenden Tonträgermarkt[34] und profitierten davon gemeinsam mit allen anderen Marktteilnehmern .
- Dass ein Standard eine großartige Werbung für ein Unternehmen sein kann, hat Sun mit der Einführung des Java-Standards bewiesen. Sie profitierten vor allem durch einen Imagegewinn und weniger durch direkte Einnahmen. Diese Strategie wird auch oft mit der Marktbelegung kombiniert. Firmen erhöhen das Marktvolumen durch einen offenen Standard und versuchen gleichzeitig den Standard mit ihrem Firmennamen zu verbinden, um sich innerhalb des vergrößerten Marktes auch noch einen höheren Marktanteil zu sichern (siehe Kapitel 5.4).

- Eine weitere Möglichkeit den eigenen Marktanteil an Produkten des Standards zu erhöhen ist den Know-How Vorteil aus der Entwicklung des Standards zu nützen. Das Unternehmen, das den Standard verwaltet, kennt neue Versionen als erstes und muss je nach Grad der Offenheit auch nicht alle Details der Spezifikation veröffentlichen.
- Eine ganz andere Form von Standards zu profitieren haben wir unter dem Begriff „direkte Erlöse“ zusammengefasst. Das reicht von Lizenzeinnahmen bis zu Transaktionsgebühren, wie sie zum Beispiel von electronic payment Anbietern eingehoben werden.
- Vom Lock-In von Anwendern profitieren Unternehmen, wenn diese Anwender weitere Investitionen in Produkte des Unternehmens tätigen müssen. Sie sind durch switching costs gezwungen Komplementärprodukte, Verbrauchsmaterialien oder Nachfolgeprodukte bei diesem Unternehmen zu kaufen. (siehe Kapitel 6.2)

Zusätzlich zu diesen Elementen einer Finanzierung können bei der Anwendung des Frameworks noch weitere kreative Möglichkeiten entwickelt werden, wie der Standard zu nützen ist. Viele Strategien, wie zum Beispiel PayPals anfängliche Absicht sich von den Zinsen des bei ihnen zwischengelagerten Geldes zu finanzieren, lassen sich keiner dieser Kategorien zuordnen.

Nun gilt es zu beurteilen, welche Formen der Finanzierung möglich sind und wie viel Profit sie einzubringen versprechen. Nach unserem Framework werden die Möglichkeiten erstens einmal direkt durch die Faktoren Ausschließbarkeit und Konkurrenz eingeschränkt. Wenn Unternehmen über keine Mittel verfügen Konkurrenten auszuschließen, so werden direkte Finanzierungsmethoden, wie Lock-In oder direkte Erlöse, entscheidend geschwächt, da der Markt allen anderen Unternehmen genauso offen steht. Wenn ein Standard public good ist, können zum Beispiel keine Lizenzgebühren dafür verlangt werden. Ebenso sind die meisten Einnahmequellen umso weniger profitabel, je stärker die Konkurrenz ist.

Noch wichtiger ist allerdings der indirekte Einfluss von Faktoren wie excess inertia oder dem Nutzen von Mix und Match. Diese Faktoren wirken in unserem Framework auf die Offenheits- und/oder Kompatibilitätsstrategie ein. Sie bestimmen also, welches Maß der Offenheit oder der Kompatibilität optimal für die Einführung des Standards ist. Nachdem diese beiden Strategien bestimmt wurden, werden sie auf die möglichen Finanzierungsstrategien abgestimmt. Das heißt es wird überprüft, welche Möglichkeiten der Finanzierung mit welchem Ausmaß von Offenheit oder Kompatibilität vereinbar sind. Dann werden Profitvorteile eines proprietären Standards gegenüber Netzwerkvorteilen eines offeneren Standards abgewogen.

Die **Offenheitsstrategie** besteht aus den drei Skalen, die im Kapitel 7.1 definiert wurden: Offenheit der Verwendung, der Entwicklung und der Anpassbarkeit des Standards. Für das Entwickeln einer grundlegenden Strategie sind vor allem die ersten beiden Skalen von Bedeutung. Durch Offenheit in diesen beiden Dimensionen wird ein Standard konkurrenzfähiger und die Gefahr von excess inertia wird reduziert. Ein Standard, der für alle frei zu verwenden ist, reduziert Einnahmen aus direkten Erlösen und dem Lock-In von Kunden, erhöht aber die Gewinne aus einer Belegung des Marktes. Wenn auch die Entwicklung des Standards geöffnet wird, werden der Know-How Vorsprung und manchmal auch der Imagegewinn reduziert, etwa bei der Entwicklung durch eine unabhängige Standardisierungsorganisation.

Die optimale Offenheitsstrategie ist abhängig von den folgenden Faktoren:

- **Gefahr von excess inertia:** Offenheit kann ein geeignetes Mittel sein Anwender anzuziehen oder eine größere Anzahl von potentiellen Anwendern anzusprechen und dadurch das Netzwerk rasch zu vergrößern.

- **Angst vor Lock-In:** Durch Offenheit kann Anwendern garantiert werden, dass der Lock-In von Anwendern nicht ausgenutzt wird. Da auch andere Unternehmen den Standard unterstützen können sind Anwender nicht einem einzelnen Unternehmen ausgeliefert.
- **Konkurrenz:** Je nachdem, welche Konkurrenten einem gegenüberstehen, kann man unterschiedliche Grade der Offenheit bevorzugen. Offenheit kann eine Möglichkeit sein gegen übermächtige Konkurrenten anzutreten. So hat zum Beispiel Netscape Microsofts Übermacht eine radikale Öffnung seines Browsers entgegengesetzt. Eine andere mögliche Überlegung ist es, die Ebene des Standards, auf der starke Konkurrenz zu erwarten ist, zu öffnen, um Netzwerkexternalitäten zu gewinnen und auf Ebenen mit schwächerer Konkurrenz selbst Profite zu erwirtschaften.
- **Ausschließbarkeit:** Ausschließbarkeit bestimmt das Ausmaß, in dem ein Standard überhaupt proprietär gehalten werden kann. Im Extremfall hat ein Unternehmen keinerlei Möglichkeiten andere von der Verwendung des Standards auszuschließen und wird so zu einer Offenheits-Strategie gezwungen.

Es ist allerdings wichtig zu verstehen, dass dieser Begriff der Offenheit und die Einflüsse der Faktoren Verallgemeinerungen sind, die zwar tendenziell aber nicht immer gelten. So kann zum Beispiel der positive Effekt von Offenheit auf die Verbreitung des Standards auch sehr gering ausfallen. Ein gutes Beispiel für eine fein abgestimmte Offenheitsstrategie bietet wieder der PDF-Standard. Der Standard wird vor allem von Firmen oder Organisationen verwendet, die ihre Dokumente einer großen Anzahl von Anwendern zur Verfügung stellen wollen und deutlich weniger zum Austausch von Informationen zwischen einzelnen Personen. Hätte Adobe den Acrobat-Reader nicht gratis zur Verfügung gestellt, so hätte sich das PDF-Format aufgrund von excess inertia wohl kaum durchsetzen können. Kein Anwender wäre bereit gewesen für ein Programm zu bezahlen, nur um Dateien eines bestimmten Formats anzeigen zu können, das noch dazu zu Beginn nur wenig verbreitet war.

Doch auch die Software zum Erstellen von PDF-Files kostenlos anzubieten hätte Adobe vergleichsweise wenig Netzwerkvorteil eingebracht. Denn jene Unternehmen und Organisationen, die PDF-Files erstellen, sind auch bereit dafür Lizenzgebühren zu bezahlen, solange sie damit eine große Anzahl an Anwendern erreichen. Da also wenige Anwender PDF-Dokumente erstellen, aber viele Anwender diese lesen sollen, genügt Offenheit auf Ebene der Reader-Software.

Außerdem soll festgehalten werden, dass ein offener Standard keineswegs gleich bedeutend mit weniger Gewinn ist. Oft genug ist genau das Gegenteil der Fall. Offenheit und Kompatibilität reduzieren lediglich manche Formen der direkten Finanzierung, wie Lizenzeinnahmen. Dafür ermöglichen sie oft erst eine starke Verbreitung des Standards und damit verbundenen Profit aus Imagegewinnen und Erhöhung der Konsumentennachfrage.

Mit der Offenheitsstrategie eng verbunden ist die Entscheidung über **Kompatibilität und Kooperation**. Wenn ein Standard gemeinsam mit anderen Unternehmen eingeführt wird, muss automatisch ein gewisses Maß an proprietärer Kontrolle aufgegeben werden. Und wenn ein Standard allen offen zur Verfügung gestellt wird, kann die Unterstützung durch Kooperationspartner nur von Vorteil sein, da bei einer Offenheitsstrategie kaum eigene Gewinne für die Kooperation geopfert werden müssen.

Typische Formen von Kooperationen und Kompatibilität wurden in Kapitel 8.1 beschrieben. Unternehmen können sich zum Beispiel darauf einigen einen gemeinsamen Standard zu adaptieren, ihre Netzwerke zusammenschließen oder Konvertierungsprogramme zur Verfügung stellen.

Vor allem im Zusammenhang mit der Finanzierung ist es wesentlich, ob es sich um horizontale oder vertikale Kooperationen handelt, inwiefern also der Kooperationspartner ein direkter Konkurrent ist.

Die folgenden Faktoren wirken auf die Kompatibilitätsentscheidung ein:

- **Gefahr von excess inertia:** Excess inertia kann durch Kooperation sehr direkt unterbunden werden, indem zwei Unternehmen ihre Netzwerke verbinden und so die kritische Masse erreichen. Indem ein neuer Standard zu einem bestehenden System kompatibel gestaltet wird, können die Netzwerkeexternalitäten des alten Netzwerkes genutzt werden. Anwender zögern nicht den neuen Standard zu adaptieren, da sie mit Anwendern des bestehenden Standards kommunizieren können.
- **Bestehende switching costs:** Ebenso werden switching costs durch Kompatibilität gesenkt. Wenn eine große Anzahl von Anwendern einem konkurrierenden Netzwerk angehört und dieses Netzwerk aufgrund bestehender switching costs nicht verlassen kann, so ist es sinnvoll diesen Anwendern zum Beispiel mit Adaptern den Umstieg zu erleichtern.
- **Konkurrenz:** Die Konkurrenz beeinflusst Kooperation auf dreifache Weise. Erstens wird die Kooperation unter Umständen mit Konkurrenzunternehmen eingegangen. Dafür müssen die Stärken und Schwächen der Konkurrenten beurteilt werden, um einerseits zu wissen wer ein lohnender Kooperationspartner ist und andererseits wer als Konkurrent zu mächtig ist und daher besser zur Teilnahme an der Kooperation bewegt werden soll. Zweitens sind Kooperationen mehrerer kleiner Unternehmen ein Mittel sich gegen übermächtige Konkurrenten zu behaupten. Drittens verlagern Kooperationen, wie in Kapitel 8.2 besprochen wurde, den Wettbewerb von Systemen zu Komponenten und verändern den Grad des Preiswettbewerbs. Abhängig von den Stärken und Schwächen der Konkurrenz und ihrer Produkte können diese Veränderungen mehr oder weniger erwünscht sein.
- **Ausschließbarkeit:** Wenn Konkurrenten von einem Standard ohnehin nicht ausgeschlossen werden können, verliert ein Unternehmen wenig, wenn es gleich eine Kooperation eingeht und sich so die volle Unterstützung sichert.
- **Angst vor Lock-In:** Genauso wie durch Offenheit wird auch durch Kooperationen die Angst der Konsumenten vor Lock-In reduziert.
- **Nutzen von „Mix and Match“:** Je höher der Nutzen von „Mix and Match“ für Konsumenten, umso höher ist auch der Anreiz für Unternehmen Kompatibilität herzustellen, um mehr Anwender bedienen zu können.

Die drei Strategiekomponenten Finanzierung, Offenheit und Kooperation werden zu eine Gesamtstrategie zusammengefügt. Dabei muss nicht nur betrachtet werden, welche Formen der Offenheit und Kooperation, welche Finanzierungsmöglichkeiten zulassen, sondern auch wie diese beiden Strategien die Rahmenbedingungen des Wettbewerbs verändern. So kann zum Beispiel ein gemeinsamer Standard den Spielraum für Innovationen reduzieren. Im Zusammenhang mit der Finanzierungsstrategie muss beurteilt werden, ob das erwünscht, tragbar oder untragbar ist.

Wurde eine Gesamtstrategie festgelegt, so kann man zur nächsten Phase der Feinabstimmung übergehen.

9.4 Phase 3 - Feinabstimmung der Strategie

In der Feinabstimmung werden in neun Schritten weitere Details der Einführungsstrategie ausgearbeitet. Nicht jeder dieser Schritte ist für jede Situation relevant. So kann zum Beispiel der Zeitpunkt der Einführung in Situationen ohne first-mover advantage allein durch Überlegungen der Produktion und des Marketings bestimmt werden.

Die Schritte der Feinabstimmung werden durch verschiedene Faktoren beeinflusst, die in Abbildung 9.1 durch eine Schnittstelle zwischen der zweiten und der dritten Phase symbolisiert sind. Die Faktoren der Schnittstelle beschreiben die Situation nach Anwendung der Grundstrategie.

Die folgenden Faktoren sind für die Feinabstimmung der Strategie relevant:

- **Gefahr von excess inertia:** Dieser Faktor gibt an, ob die Gefahr von excess inertia durch die Maßnahmen der Grundstrategie schon ausreichend reduziert wurde oder ob weitere Maßnahmen nötig sind.
- **Gefahr der Fragmentierung:** Vor allem durch Offenheit in der Entwicklung und hohe Anpassbarkeit des Standards kann die Gefahr der Fragmentierung erhöht werden. (siehe Kapitel 7.1)
- **Ausschließbarkeit:** Die Ausschließbarkeit beschreibt wieder, ob Unternehmen über Maßnahmen verfügen Konkurrenten an der Nutzung des Standards zu hindern. Dieser Faktor wird einerseits von den grundlegenden Möglichkeiten der Ausschließbarkeit beeinflusst, die in der ersten Phase definiert wurden. Andererseits beeinflusst aber auch der gewählte Grad der Offenheit diesen Faktor, da eine Verpflichtung zu Offenheit oft nicht erlaubt Konkurrenten auszuschließen.
- **Grundstrategie:** Die Grundstrategie besteht aus der gewählten Kombination von Offenheit, Kooperation und Finanzierung.
- **First-mover advantage:** Dieser Faktor gibt an, wie wichtig ein first-mover advantage für die Einführung des Standards ist und ob die Konkurrenz oder das eigene Unternehmen einen solchen Vorteil nützen kann.
- **Switching costs:** Switching costs haben für die Feinabstimmung eine mehrfache Bedeutung, da sowohl die switching costs zum eigenen Standard hin als auch vom eigenen Standard weg beachtet werden müssen. Besonders wichtig ist es switching costs aufzubauen, wenn „Lock-In der Anwender“ ein Teil der Grundstrategie ist.
- **Form der Kooperation:** Je nach Kooperationsstrategie werden unterschiedliche Formen der Kooperation gewählt und ist es unterschiedlich schwer Partner für die Kooperation zu gewinnen.

Diese Faktoren bestimmen die Bedeutung der folgenden Maßnahmen der Feinabstimmung. Jede der angeführten Maßnahmen setzt sich aus mehreren Entscheidungen zusammen, die alle in dieser Arbeit besprochen wurden:

- **Vorankündigungen und Konsumentenerwartungen:** Wie im Kapitel 4 diskutiert, ist oft für die Verbreitung eines Standards nicht die tatsächliche sondern die erwartete Größe des Netzwerks ausschlaggebend. Deswegen bemühen sich Unternehmen das Vertrauen von Kunden in ihren Standard zu erhöhen, um excess inertia zu verhindern. Das geschieht

meist durch traditionelle Marketingmaßnahmen, allerdings nicht mit dem Ziel Vorzüge des Produktes zu bewerben, sondern um zu zeigen, wie viele Anwender und Unternehmen den Standard bereits unterstützen. So warben zum Beispiel sowohl U.S. Robotics als auch Rockwell im Kampf um den 56K-Modem Standard damit, dass der Großteil der Internet Service Provider ihren Standard unterstützte[48]. Eine besondere Rolle spielen Produktvorankündigungen, da sie zusätzlich zur Werbewirkung auch den first-mover advantage verschieben können.(siehe Kapitel 4.3)

- **Verhindern von Fragmentierung:** Entsteht durch Offenheit die Gefahr der Fragmentierung, so müssen nun Maßnahmen entwickelt werden, um diese Gefahr zu reduzieren. Der erste Schritt ist beim Design des Standards darauf zu achten, uneindeutige Elemente oder zu viel Interpretationsspielraum zu vermeiden. Auch eine ausführliche Dokumentation hilft zumindest unbeabsichtigte Fragmentierung zu reduzieren. Eine weitere Möglichkeit ist eine Compliance-Marke einzuführen, die nur Produkte tragen dürfen, die mit dem Standard voll kompatibel sind. Sun hat zum Beispiel die Lizenzbedingungen für Java so gestaltet, dass Fragmentierung minimiert wird. Unternehmen, die in ihren Produkten eine Java Virtual Machine verwenden wollen, werden in der Lizenz verpflichtet die JVM standardkonform zu implementieren. Wegen Verletzung dieser Lizenzbedingung verklagte Sun Microsoft. Auch die (kostenlose) Lizenz für Softwareentwickler enthält eine Klausel, dass die Standard-Java Klassen (java.*) nicht verändert werden dürfen.

Fragmentierung entsteht oft aus dem Bedürfnis sich von Konkurrenten zu differenzieren. Durch den Standard sind Differenzierungsmöglichkeiten eingeschränkt und daher versuchen Unternehmen den Standard zu erweitern. Dem entgegen wirken starke Netzwerkexternalitäten. Denn wenn ein Unternehmen nicht selbst einen großen Teil der Anwender des Standards kontrolliert, verliert es durch teilweise Inkompatibilität Netzwerkexternalitäten.

- **Ausmaß der Netzwerkexternalitäten managen:** Dieser Punkt bestimmt alle Entscheidungen, die das Ausmaß der Netzwerkexternalitäten und die angestrebte Reichweite des Netzwerks angeben. Das Ausmaß der Netzwerkexternalitäten eines Standards ist nicht unbedingt vordefiniert, sondern hängt von der Detailimplementation ab. Im Kapitel 4.1 wurde als hypothetisches Beispiel beschrieben, wie der Rezeptstandard auch ohne Netzwerkfaktor konstruiert werden könnte. Abhängig von der Grundstrategie und der Gefahr von excess inertia kann es sinnvoll sein den Netzwerkfaktor am Anfang zu reduzieren und erst wenn ein ausreichendes Netzwerk aufgebaut wurde einzusetzen, wie im Kapitel 4.1 am Beispiel von amazon.com gezeigt wurde. Außerdem gilt es zu beurteilen, ob Netzwerkexternalitäten auf Regionen, Sprachen oder andere Anwendergruppen beschränkt sind. In diesem Fall kann es sich empfehlen die Technologie zuerst in einem kleineren Kreis einzuführen, in dem dafür Netzwerkexternalitäten umso stärker sind. Ein Online-Forum über Fernsehsendungen hat zum Beispiel mehr Nutzen, wenn 100 Seher des gleichen Senders miteinander diskutieren, als wenn die 100 Teilnehmer über die ganze Welt verstreut sind. Wenn die Marketingmaßnahmen lokal konzentriert werden, können also mit dem gleichen Aufwand höhere Netzwerkexternalitäten erreicht werden.
- **Maßnahmen gegen excess inertia:** In diesem Schritt werden, wenn nötig, über die Offenheits- und Kooperationsstrategie hinausgehende Methoden zur Vermeidung von excess inertia entwickelt. Die beiden prinzipiellen Ansätze sind, den stand-alone value zu erhöhen oder möglichst rasch ein großes Netzwerk aufzubauen. Diese Strategien müssen auf den einzelnen Fall abgestimmt werden und lassen sich schwer kategorisieren. Eines der

Beispiele in dieser Arbeit ist das „Schneeballsystem“ von PayPal, das es erlaubt auch noch nicht angemeldeten Usern mit PayPal Geld zu schicken. Sony und Philips haben sich bei der Einführung der CD zu Beginn an die High-End Anwender gerichtet, da für sie die CD-Technologie einen höheren stand-alone value bot. Online Communities werden oft mit „Big Bangs“ eröffnet, also Marketingmaßnahmen, die auf einen kurzen Zeitraum konzentriert sind, um möglichst rasch eine kritische Masse zu erreichen.

- **Sponsorship:** Sponsorship kann eingesetzt werden um excess inertia zu verhindern, eine installed base von gebundenen Usern aufzubauen oder Anwendern die switching costs beim Wechsel zum eigenen Standard abzunehmen. Sponsorship macht vor allem dann Sinn, wenn Konkurrenten von der aufgebauten installed base ausgeschlossen werden können. Denn sonst profitieren die Konkurrenten von den sponsorship-Maßnahmen genauso wie das eigene Unternehmen. Druckerhersteller versuchen zum Beispiel zu verhindern, dass andere Anbieter Nachfülltinte für ihre Drucker anbieten. Denn der Hersteller des Druckers bietet seine Geräte billiger an, um eine installed base an Kunden aufzubauen, die seine teureren Tintenpatronen kaufen müssen. Wenn nun andere Unternehmen Nachfülltinte anbieten, verliert der Produzent der Tinte seine sponsorship-Investitionen. Besonders gut für sponsorship geeignet sind einmalige oder zeitlich begrenzte Marketingaktionen, wie zum Beispiel Gratishandys bei der Erstanmeldung oder Prämien für die Werbung von Neukunden.
- **Zeitplan:** Je stärker die Netzwerkexternalitäten sind und je schneller sich ein Markt entwickelt, umso größer wird die Bedeutung eines Zeitvorsprungs bei der Einführung. Durch first-mover advantages ist es wichtig seine Technologie als erster einzuführen oder den Konkurrenten zumindest nicht zu viel Zeitvorsprung zu gewähren. Auf der anderen Seite schadet die verfrühte Einführung einer unreifen Technologie dem wichtigen Konsumentenvertrauen. Auch wenn die Technologie eingeführt wird, bevor man sich die Unterstützung gewünschter Partner gesichert hat, kann die Einführung an zu wenig Vertrauen der Konsumenten scheitern. Neben der Frage des richtigen Zeitpunktes können durch die Grundstrategie noch weitere Timing-Entscheidungen nötig werden. So kann bei einem offenen Standard der Zeitvorsprung, den die eigene Entwicklungsabteilung gegenüber der Konkurrenz hat, der wichtigste Wettbewerbsvorteil sein.
- **Managen der switching costs:** Zunächst muss das optimale Ausmaß der switching costs bestimmt werden. Switching costs von einem alten zum eigenen Standard und switching costs, wenn Anwender den eigenen Standard wieder verlassen korrelieren oft. Je nach Grundstrategie kann es entweder wichtiger sein Anwendern einen reibungslosen Umstieg zum eigenen Standard zu ermöglichen oder sie möglichst eng an den Standard zu binden. Ersteres wird vor allem durch Kompatibilität und sponsorship erreicht. Für zweiteres wurden in Kapitel 6.2 verschiedene Maßnahmen diskutiert, wie Anwender dazu gebracht werden können ihre eigenen switching costs zu erhöhen. Eine weitere Überlegung ist es den Standard so zu designen, dass neue User oder User mit niedrigen switching costs von bereits gebundenen Anwendern unterschieden werden können. Dadurch wird gezielteres Sponsoring möglich.
- **Free-ridership verhindern oder nützen:** Je weniger Möglichkeiten ein Unternehmen hat Konkurrenten vom Standard auszuschließen, umso größer ist die Gefahr, dass sich andere Unternehmen nicht an der Entwicklung des Standards beteiligen ihn aber trotzdem nützen. Durch Ausschöpfen der verschiedenen Maßnahmen des Ausschlusses (siehe

Kapitel 5.2) kann versucht werden diese free-ridership zu minimieren. Meist ist aber die erfolgreiche Entwicklung des Standards wichtiger als free-ridership durch Konkurrenten zu verhindern. Deswegen wird diese Entscheidung erst in der dritten Phase getroffen, nachdem schon Grad der Offenheit und Kooperationsform bestimmt wurden. Wenn keine tauglichen Mittel zum Ausschluss von Konkurrenten vorhanden sind, kann es sinnvoll sein free-rider in Bereichen, an denen das eigene Unternehmen wenig Interesse hat zu unterstützen. So hat zum Beispiel die Offenheit des PDF-Standards einen PDF-Konverter für LaTeX-Dokumente ermöglicht. Da in der LaTeX Umgebung die meisten Programme kostenlos zur Verfügung gestellt werden, hat Adobe dadurch keine Einnahmen verloren, aber einen Netzwerkvorteil gewonnen.

- **Kooperation aufbauen:** Auch für den Aufbau einer Kooperation ist es wichtig free-ridership zu überwinden. Den potentiellen Partnern muss klar gemacht werden, welche Vorteile eine aktive Beteiligung an der Kooperation bietet (siehe Kapitel 5.3). Speziell wenn Unternehmen einen Standard jederzeit adaptieren können, kann es für sie Sinn machen abzuwarten, wie sich der Standard entwickelt und ihn erst zu unterstützen, wenn er sich als erfolgreich herausstellt. Wie in Kapitel 8.3 beschrieben, liegt es nicht immer im Interesse aller Unternehmen einen Standard zu adaptieren. Unternehmen können unterschiedliche Standards bevorzugen oder eines kann überhaupt nicht an einer Standardisierung interessiert sein. Deshalb ist es zuallererst notwendig die eigenen und die fremden Präferenzen nach der in Kapitel 8.3 präsentierten Payoff-Matrix einzuschätzen. Dann können strategische Maßnahmen ergriffen werden um das gewünschte Ergebnis zu erreichen. Die Parameter einer Kooperation müssen sorgfältig geplant werden, sodass sie eine faire Nutzenverteilung für alle Beteiligten bietet. Einer der Gründe, warum die holländische Chipkarten-Kooperation zerbrochen ist, war nach De Vries und Hendrikse [10], dass die Gewinne aus der Kooperation nicht gerecht unter den Unternehmen verteilt waren.

Nachdem diese Schritte durchlaufen wurden, ist die Strategie für die Einführung des Standards komplett. Sie sollte als flexibles Instrument aufgefasst werden, das die strategische Richtung vorgibt, aber in Details an reale Verhältnisse angepasst werden kann.

Kapitel 10

Anwendung des Entscheidungsframeworks auf das fiktive Unternehmen ModernCooking

Zum Abschluss der Arbeit wird die Anwendung aller beschriebenen Strategieelemente und des Entscheidungsframeworks an der ausführlichen Ausarbeitung eines Beispiels demonstriert. Anhand des Frameworks wird eine Einführungsstrategie für den in Kapitel 3 beschriebenen Rezeptstandard erstellt.

Für die Ausarbeitung dieses fiktiven Beispiels müssen noch einige Voraussetzungen festgelegt werden. Das Unternehmen ModernCooking ist ein mittelgroßes Unternehmen, das bereits seit einigen Jahren mit Produkten am Markt vertreten ist. Das Unternehmen ist in der Lage die für den Standard am wichtigsten Geräte, wie Kühlschrank, Herd und einige Kleingeräte selbst herzustellen, bietet aber nur eine relativ kleine Auswahl und wenig darüber hinausgehende Produkte an. Die Küchengeräte sind über ein Netzwerk miteinander und mit einer Zentraleinheit verbunden, die für Internetzugang, Druckfunktion und Speicherung der Daten verantwortlich ist.

Jedes der folgenden drei Kapitel widmet sich einer Phase des Frameworks. In Kapitel 10.4 werden dann die Ergebnisse zusammengefasst und kurz mit Standardisierungen im realen Haushaltsgerätemarkt verglichen.

10.1 Phase 1 - Marktanalyse

In der ersten Phase der Marktanalyse werden die Einflussfaktoren des Frameworks Schritt für Schritt ermittelt.

Stand-alone value

Bei der Beurteilung der Eigenschaften des Standards muss zwischen den Eigenschaften der Küchengeräte und den Eigenschaften des Rezeptstandards selbst unterschieden werden. Beide sind für unsere Überlegungen relevant. Die Eigenschaften der Küchengeräte müssen miteinbezogen werden, da sie für die Kaufentscheidung von Konsumenten eine wichtige Rolle spielen und damit die Verbreitung von CookML-fähigen Küchengeräten mitbestimmen. Allerdings würde es zu kurz greifen nur die Küchengeräte als Ganzes zu beurteilen, da das Ziel ja ist eine hohe Verwendung des Standards zu erreichen. Die Verbreitung CookML-fähiger Küchengeräte garantiert noch nicht, dass Anwender CookML-Rezepte erstellen oder dass Rezeptdatenbanken Downloads im CookML Format anbieten.

Der stand-alone value der Küchengeräte ist sehr hoch. Auch ohne CookML-Netzwerk können die meisten Funktionen genutzt werden, denn ein Kühlschrank kühlt auch wenn er der einzige seiner Art ist. Die Zentraleinheit hat schon einen geringeren stand-alone value, da der Großteil ihrer Aufgaben, wie Verbindung der Geräte, Rezeptdownload und Erstellung einer Einkaufsliste, auf den Rezeptstandard angewiesen ist. Doch auch ohne Rezepte kann sie einige Funktionen, wie Inventarisierung des Kühlschranks, Online-Bestellung oder Online-Wartung der Geräte erfüllen.

Die Rezeptverarbeitungsfunktion selbst bietet einen mittleren stand-alone value, da auch sie ohne Netzwerk eingeschränkt genutzt werden kann. Anwender können ihre eigenen Rezepte eingeben und Küchengeräte werden wahrscheinlich mit einer Grundausstattung an Rezepten geliefert werden. Doch das wahre Potential mit CookML-Schnittstellen zu Rezeptdatenbanken, Ernährungsberatern und Supermärkten kann die Rezeptfunktion erst erreichen, wenn ein ausreichendes Netzwerk aufgebaut ist.

Netzwerkexternalitäten

Wie in Kapitel 4.1 hergeleitet, treten beim Rezeptstandard sowohl direkte als auch indirekte Netzwerkexternalitäten auf.

Direkte Netzwerkexternalitäten werden durch den Austausch von Rezepten zwischen Anwendern erzeugt. Der Austausch kann zum Beispiel über Rezeptdatenbanken im Internet, über private Homepages oder einfach per E-Mail erfolgen. Je mehr Anwender den Rezeptstandard verwenden, umso mehr Rezepte stehen Anwendern zum Tausch zur Verfügung. Für die Funktion des Rezeptaustausches sind Netzwerkexternalitäten mittelmäßig stark. Denn es wird zwar ein Netzwerk von einer gewissen Größe benötigt, um eine ausreichende Auswahl an Rezepten zu ermöglichen. Doch Netzwerkexternalitäten sind nicht so stark, dass der Standard erst nutzbar ist, wenn der Großteil der Menschen diesem Netzwerk angehört. Zum Beispiel bei einem Telefonnetzwerk ist es wichtig, dass jede Person, die man erreichen will dem selben Netzwerk angehört. Deshalb sind alle lokalen Telefonnetzwerke zu einem weltweiten Netz zusammengeschlossen. Beim CookML-Standard ist es hingegen weniger wichtig, dass mit einer bestimmten Person Rezepte ausgetauscht werden können, als dass im Internet genügend Rezepte von beliebigen Personen zur Verfügung stehen.

Indirekte Netzwerkexternalitäten entstehen durch die Komplementarität zwischen Rezepten auf der einen und Küchengeräten, Kochsoftware, Rezeptdatenbanken oder Kochbüchern auf der anderen Seite. Je mehr Anwender den CookML-Standard verwenden, umso mehr kompatible Güter werden zur Verfügung gestellt. Diese Netzwerkexternalitäten können um eine Spur stärker eingeschätzt werden als direkte Netzwerkexternalitäten. Denn die indirekten Netzwerkexternalitäten erhöhen das Bedürfnis nach einem *einzigem* Standard. Für den Rezeptaustausch ist es weniger wichtig, wie viele Rezeptstandards es gibt, solange jeder ein ausreichend großes Netzwerk hat. Anwender tauschen im Internet dann einfach nur mit jenen Anwendern, die den gleichen Standard verwenden. Für diejenigen, die Produkte für einen Standard entwickeln, ist es allerdings hinderlich, wenn sie nur die Hälfte der Anwender bedienen können oder Funktionalität für zwei Standards entwickeln müssen.

Eigene Marktmacht

Wie in der Einleitung definiert, verfügt ModernCooking nur über geringe Marktmacht. Sie sind mittelmäßig bekannt und haben noch keinen nennenswerten Kundenstamm.

Die installed base spielt für diesen Standard keine Rolle, da vollkommen neue Produkte eingeführt werden und sich alte Küchengeräte nicht aufrüsten lassen. Daher hat ModernCooking keinen direkten Nachteil durch ihren geringen Marktanteil, sondern nur einen indirekten durch ihren geringeren Bekanntheitsgrad.

Beurteilung der Gefahr von excess inertia

Ein großer Vorteil des Standards ist der hohe stand-alone value der Küchengeräte. Das gekaufte Produkt bietet dem Kunden dadurch schon Nutzen, bevor ein Netzwerk aufgebaut ist. Wenn das Netzwerk der Anwender mit CookML-Geräten schließlich groß genug ist, können sie beginnen Rezepte herunterzuladen und auszutauschen. Durch die Verknüpfung mit Küchengeräten mit hohem stand-alone value wird also excess inertia reduziert, allerdings mit den folgenden Einschränkungen:

1. Auch wenn das Netzwerk der CookML-Geräte groß genug ist, müssen Anwender erst anfangen den Standard zu nützen.

2. Solange der CookML Standard nicht genutzt wird, bieten CookML-taugliche Geräte den Anwendern wenig Mehrwert. Daher sind Anwender auch nicht bereit für solche Geräte mehr zu bezahlen beziehungsweise wird die Nachfrage durch diese Geräte nicht steigen.
3. Wenn Anwender die CookML-Funktionen zu Beginn nicht nützen können, so kann das negative Auswirkungen auf das Image des Standards haben, die später nur mehr schwer zu korrigieren sind.

Bezieht man auch den vorhandenen stand-alone value des Rezeptstandards und die mittelstarken Netzwerkexternalitäten in die Überlegung mit ein, so scheint die Gefahr von excess inertia durchaus gegeben, aber nicht unüberwindbar. Wenn nur vereinzelte Unternehmen CookML in ihren Produkten unterstützen und nur wenige Anwender ihre Rezepte im CookML Format anbieten, so wird der Rezeptstandard nie eine umfassende Küchenhilfe darstellen können. Doch die Schwelle der kritischen Masse scheint nicht allzu hoch und mit geschickten Maßnahmen wird sich excess inertia überwinden lassen. Die größte Schwachstelle ist ModernCookings geringer Bekanntheitsgrad, der die Schwierigkeit der Einführung deutlich erhöht.

Ausschließbarkeit

Es bieten sich nur wenige Möglichkeiten Konkurrenten vom Rezeptstandard auszuschließen, ohne die einfache Verwendbarkeit und das Austauschen von Rezepten über verschiedenste Medien einzuschränken. Da eine große Anzahl von Unternehmen CookML Rezepte in ihren Produkten verwenden soll, muss die Spezifikation des Standards öffentlich bekannt sein. Es ist relativ leicht möglich einen Interpreter für Kochrezepte zu entwickeln, sodass Konkurrenten auch nicht durch Komplexität ausgeschlossen werden können.

Es bleiben also nur noch rechtliche Mittel. Eine Möglichkeit ist den Standard patentieren zu lassen und von Unternehmen, die den Rezeptstandard in ihren Produkten verwenden wollen, Lizenzgebühren einzuheben. Allerdings ist nicht nur die Patentierbarkeit des Rezeptstandards fraglich, sondern auch ob ein solches Verbot, ein bestimmtes Format zu interpretieren, weltweit durchsetzbar wäre. Selbst dann könnten Firmen noch immer strukturell ähnliche Standards erstellen, die automatisch nach CookML konvertierbar wären. Diese Lösung erscheint daher eher unpraktikabel.

Durchaus machbar wäre hingegen die Einführung einer Compliance-Marke. Firmen können ihre Produkte auf Kompatibilität mit dem CookML Standard überprüfen lassen und dürfen erst nach bestandener Prüfung ein „CookML Compliant“-Logo führen, das markenrechtlich geschützt ist. Je bekannter die CookML-Marke ist, umso eher werden Unternehmen versuchen ihre Produkte zertifizieren zu lassen. Für die Compliance-Prüfung können natürlich auch Gebühren eingehoben werden.

Marktmacht der Konkurrenz

Im Küchengerätemarkt sind neben mehreren großen Playern, wie Miele und Siemens, auch viele kleine spezialisierte Unternehmen, wie zum Beispiel Soehnle im Waagenbereich, tätig. Vor allem die großen Unternehmen haben große Imagevorteile gegenüber jungen, unbekannten Marken und Konsumenten würden definitiv einen Standard bevorzugen, der von großen Namen unterstützt wird.

Aber auch die großen Unternehmen verfügen über keine installed base, die sich direkt für die neue Generation der Küchengeräte nutzen ließe.

Diese Unternehmen haben bis jetzt keine Versuche unternommen einen Rezeptstandard zu entwickeln oder einzuführen. Der einzige vergleichbare Standard für Kochrezepte wurde von einem kleinen amerikanischen Unternehmen namens FormatData entwickelt. Der Standard hat den Namen RecipeML und ist in Form einer XML-DTD spezifiziert[17]. Der Standard wird von zwei Ernährungs- und Rezeptwebseiten, dem Hersteller einer professionellen Rezeptverwaltungs-Software und einem Ofen-Hersteller unterstützt. Diese Partner sind allesamt kleinere Unternehmen und verwenden RecipeML noch nicht in ihren Produkten. RecipeML hat noch keine nennenswerte installed base aufbauen können und hat nur eine sehr geringe Marktmacht.

Eintrittsbarriere

Die Eintrittsbarriere für neue Anbieter von Rezeptstandards ist äußerst gering, weil sich ein Standard für Rezepte relativ leicht entwickeln lässt. Da alle Rezeptstandards fundamentale Ähnlichkeiten aufweisen, können auch bestehende Standards als Ausgangsbasis für neue Entwicklungen herangezogen werden.

Diese niedrige Eintrittsbarriere bezieht sich aber lediglich auf die Entwicklung des Standards, für die erfolgreiche Einführung werden schon deutlich mehr Ressourcen benötigt.

First-mover advantage

Der einzige bestehende Rezeptstandard RecipeML existiert zwar schon seit Jänner 2000, hat aber noch kein Netzwerk aufgebaut und verfügt daher über keinen wesentlichen first-mover advantage.

Im Bereich der Küchengeräte spielt der first-mover advantage eine relativ geringe Rolle, da Küchengeräte langlebig sind und daher nur in großen Abständen ersetzt werden. Dadurch kann sich das Netzwerk der CookML-Geräte nur relativ langsam aufbauen und später am Markt Eintreffende Konkurrenten haben nur einen kleinen Netzwerknachteil gegenüber dem first-mover.

Das gilt allerdings nur, wenn die Küchengeräte der entscheidende Motor für die Verbreitung des Standards sind. Wenn zum Beispiel viele Anwender den Standard in Zusammenhang mit einer Kochsoftware verwenden oder Rezeptdatenbanken von sich aus auf CookML umsteigen, entwickelt sich das Netzwerk schneller, da das Netzwerkwachstum unabhängig von der langsamen Adaptionrate von Küchengeräten ist.

Performance

Der einzige Konkurrenzstandard RecipeML bietet eine geringere Funktionalität, als für CookML angestrebt wird. Der Standard ermöglicht keine vollständige Automatisierung von Kochabläufen. Zutaten und Arbeitsschritte werden lediglich verbal ohne eindeutige Identifizierung angegeben¹. Daher ist eine Verwendung von RecipeML statt der Entwicklung eines eigenen Standards keine Option.

Die Beurteilung der Performance von möglichen zukünftigen Entwicklungen von Rezeptstandards ist nicht möglich.

Konkurrenz

Die bestehende Konkurrenz ist gering. RecipeML ist der einzige bestehende Standard, der aber nur wenig Marktmacht, noch kein Netzwerk und unterlegene Funktionalität hat.

¹Diese Einschätzung wurde auf Grund der unter [17] angegebenen Spezifikation getroffen. Die Dokumentation der Spezifikation war allerdings zum Zeitpunkt der Erstellung dieser Arbeit noch unvollständig.

Die potentielle, zukünftige Konkurrenz stellt jedoch eine Gefahr dar. Es gibt einige große Unternehmen mit hohem Marktanteil, die diesen Marktanteil durch ihren hohen Bekanntheitsgrad wahrscheinlich auch bei Küchengeräten der nächsten Generation halten können. Das gilt sowohl für den Wettbewerb innerhalb des CookML-Standards als auch für den Fall, dass eines dieser Unternehmen einen konkurrierenden Standard einführt.

ModernCooking kann sich weder durch einen Zeitvorsprung noch durch Know-How der Technologie einen Vorteil sichern, da first-mover advantage und Eintrittsbarrieren gering sind.

Auch wenn bei der Analyse der Netzwerkexternalitäten festgestellt wurde, dass mehrere Standards im Markt überleben können, sind die dauerhaften Erfolgsaussichten für einen Standard, der gegen einen Konkurrenzstandard der „Großen“ antritt, gering.

Angst vor Lock-In

Für Anwender stellt Lock-In in diesem Fall eine beträchtliche Gefahr dar, vor allem wenn der Rezeptstandard nicht von allen Firmen unterstützt wird. Denn dann sind sie durch die switching costs bei zukünftigen Käufen von Küchengeräten auf die Produkte weniger Firmen eingeschränkt. Durch diese Einschränkung kann es Konsumenten passieren, dass sie bei der nächsten Kaufentscheidung eines kompatiblen Gerätes gezwungen sind einen höheren Preis zu bezahlen oder eine geringere Auswahl zu akzeptieren.

Die Höhe der switching costs hängt davon ab, wie wichtig die Rezeptfunktion für den jeweiligen Anwender ist und wie viele Rezepte im CookML Format er anlegt. Daher fürchten vor allem jene Anwender Lock-In, die vorhaben den Standard intensiv zu nutzen oder einen höheren Preis bezahlt haben, um ein Gerät mit Rezeptfunktion zu erhalten.

Nutzen von Mix&Match

Küchengeräte unterschiedlicher Hersteller unterscheiden sich in zahlreichen Leistungsmerkmalen und Funktionen. Diese Unterschiede werden durch die Erweiterung der Funktionalität von Küchengeräten sicher noch verstärkt. Die einzelnen Unternehmen produzieren also keine perfekten Substitute und viele Anwender verwenden Geräte unterschiedlicher Unternehmen. Besonders auf die geringe Auswahl an unterschiedlichen Modellen von ModernCooking eingeschränkt zu sein, würde für die Anwender einen gravierenden Nachteil bedeuten.

Mix&Match bietet den Konsumenten also einen sehr hohen Nutzen und wird von vielen als Voraussetzung betrachtet werden.

Bestehende Switching Costs

Da bis jetzt kein verbreiteter Standard existiert, gibt es keine allgemeinen switching costs. Rezeptdatenbanken müssen allerdings, wenn sie CookML beitreten wollen, ihren bestehenden Rezeptbestand aus ihrem eigenen proprietären Format nach CookML konvertieren.

Konsumenten werden warten bis ihre bestehenden Küchengeräte ausgetauscht werden müssen oder zumindest ein bestimmtes Alter erreicht haben, bevor sie neue Geräte erwerben. Allerdings handelt es sich dabei nicht um switching costs im eigentlichen Sinne, da sie durch ihre erste Kaufentscheidung für das alte Gerät nicht die Bedingung für spätere Kaufentscheidungen verändert haben.

10.2 Phase 2 - Das Grundgerüst der Strategie

Nun werden anhand der identifizierten Eigenschaften des Marktes die Grundpfeiler der Strategie, Ziele, Kooperation und Offenheit, ausgearbeitet.

Ziel und Finanzierung

Fest steht bereits, dass ModernCooking über Komplementäreffekte von dem Standard profitieren wird. Denn sie profitieren nicht von den Rezepten selbst, sondern auf der dazu komplementären Ebene der Küchengeräte. Die Nutzung der Komplementarität ist also von der Situation vorgegeben.

Das grundlegende Ziel von ModernCooking kann unter dem Begriff Marktbelebung zusammengefasst werden. Durch die neuen Funktionen soll die Nachfrage von Konsumenten nach neuen Geräten gesteigert werden und durch den Mehrwert der Geräte für den Konsumenten der Verkaufspreis erhöht werden. Außerdem erweitert der Standard die Palette der Küchengeräte. So können Anwender zum Beispiel mobile Terminals, sprechende Waagen oder kleine Gadgets, wie Kalorienanzeiger, erwerben.

Auch Steigerung des Bekanntheitsgrades und Aufbau eines Images als innovatives Unternehmen ist eine viel versprechende Strategie. Je kleiner die Koalition um den Standard ist oder je eher ModernCooking eine Führungsrolle in der Koalition übernimmt, umso mehr Imagevorteile kann ModernCooking erreichen. Dabei sind zwei grundsätzliche Strategien zu unterscheiden. ModernCooking kann entweder versuchen den Bekanntheitsgrad ihres Firmennamens zu erhöhen oder lediglich die Bekanntheit des CookML-Standards zu fördern. Durch die erste Möglichkeit wird der Imagegewinn auch auf andere Produkte von ModernCooking umgelegt. Außerdem kann so der Marktanteil innerhalb einer CookML-Koalition gesteigert werden. Durch die zweite Variante wird eher der Erfolg des gesamten Standards als nur der von ModernCooking erhöht. Diese Strategie ist allerdings leichter durchzuführen, da sie auch von Konkurrenten unterstützt wird. Für die Verbreitung des Standards und das Vertrauen der Konsumenten in den Standard ist ebenfalls die zweite Strategie vorteilhafter.

Auch Kombinationen dieser beiden Strategien sind möglich. So könnte zum Beispiel der Standard als Marke promotet, aber die Homepage von ModernCooking als Portal für den Standard gestaltet werden. Auf diesem Portal können Dienste für Anbieter von Rezeptdatenbanken, für Besucher über das WWW und für die automatische Interaktion mit Küchengeräten angeboten werden.

Nach unserer Analyse der Ausschließbarkeit bleiben zwei Möglichkeiten für direkte Erlöse: Lizenzen und Einnahmen aus Compliance-Tests. Lizenzgebühren für die Nutzung des Standards einzuheben scheint allerdings riskant. Abgesehen von der Frage der Durchsetzbarkeit fehlt ModernCooking zumindest in der Anfangsphase auch ein entscheidender Vorteil des Standards, der ModernCooking erlauben würde Lizenzgebühren einzuheben. Andere Unternehmen werden eher abwarten, ob sich der Standard durchsetzen kann oder ihren eigenen Standard entwickeln, bevor sie eine CookML-Lizenz erwerben. CookML bietet ihnen weder Netzwerkexternalitäten noch eine unnachahmbare Technologie. Lizenzgebühren würden also excess inertia verstärken und unter Umständen kaum Einnahmen einbringen, da der Standard nie adaptiert wird.

Die Compliance-Tests erscheinen schon vielversprechender, da sie freiwillig sind und auch eine Offenheitsstrategie zulassen. Unternehmen, für deren Produkte der Standard eine geringere Bedeutung hat oder die noch nicht sicher sind, ob sich der Standard durchsetzt, können auf die Compliance-Marke verzichten und den Standard kostenlos adaptieren.

Lock-In in den Rezeptstandard kann tatsächlich auftreten, da Konsumenten mehrere zueinander kompatible langlebige Güter erwerben und Informationsbestände an Rezepten aufbauen. Wie viel ModernCooking von diesem Lock-In profitieren kann, hängt vor allem davon ab, wie viele andere Unternehmen ebenfalls CookML kompatible Geräte anbieten und damit von der Offenheits- und Kooperationsstrategie. Denn Anwender lassen sich nur an den Standard und nicht an eine einzelne Firma binden. Wenn ModernCooking in seine installed base investiert, indem etwa Einsteigergeräte billiger angeboten werden, profitieren alle Unternehmen, die den Standard unterstützen, gleichermaßen von den neuen, gebundenen Kunden.

Von Know-How Vorteilen lässt sich in dieser Situation nur wenig profitieren, da die Komplexität des Standards gering ist, bei Updates keine schwerwiegenden Änderungen auftreten und der first-mover advantage schwach ist.

Welche sonstigen Möglichkeiten zur Finanzierung lassen sich finden? Vor allem das oben erwähnte Portal lässt sich noch weiter entwickeln. Das Portal könnte zum Beispiel als CookML-unterstützter Marktplatz für professionelle Services genutzt werden. So könnten CookML-taugliche Küchengeräte in Restaurants abhängig von der Anzahl der verkauften Speisen, anhand der Rezeptangaben, über das Portal automatisch bei Lieferanten Bestellungen aufgeben. Eine weitere Möglichkeit ist Interaktion von Privatanwendern mit Supermärkten über das Portal abzuwickeln. Anwender könnten bei Supermärkten online bestellen oder sich von Supermärkten ein „Rezept der Woche“ mit verbilligten Zutaten anzeigen lassen. Küchengeräte könnten aus der Datenbank der Supermärkte herunterladen, welche Artikel der Anwender eingekauft hat und automatisch in das Inventar aufnehmen.

Für diese Leistungen des Portals können von professionellen Anwendern, wie Restaurants, Lieferanten und Supermärkten, Gebühren eingehoben werden. Außerdem steigern sie den Nutzen von CookML und erhöhen die Bekanntheit von ModernCooking.

Zusammenfassend wurden also die folgenden Finanzierungsmöglichkeiten identifiziert:

- Marktbelebung
- Image des Standards und des Unternehmens
- Einführung einer Compliance-Marke
- Lock-In
- Einnahmen über ein Portal

Der nächste Schritt ist, zu diesen Finanzierungsmaßnahmen passende Offenheits- und Kooperationsstrategien zu entwickeln.

Kooperation

Durch die Angst der Konsumenten vor Lock-In in ein zu kleines Netzwerk und dem hohen Nutzen von Mix&Match ist klar, dass irgendeine Form der Kooperation gefunden werden muss. Diese beiden Faktoren würden zu einer sehr zögerlichen Adaption durch die Konsumenten führen und zusammen mit der identifizierten Gefahr von excess inertia den Standard scheitern lassen.

Es bleibt also zu überlegen, welche Form der Kooperation eingegangen werden soll. Dabei ist es vor allem kritisch das Ausmaß der horizontalen Kooperation zu bestimmen. Vertikale Kooperationen mit Unternehmen, wie Rezeptdatenbanken oder Softwareherstellern, werden später als zusätzliche Maßnahmen zur Verbreitung des Standards behandelt, schränken ModernCooking

aber kaum in seinen Finanzierungsmöglichkeiten ein. Es gilt also zuerst zu klären, auf welche Weise andere Küchengerätehersteller am CookML Standard beteiligt werden sollen.

Unter Berücksichtigung der Ausschließbarkeit sind die folgenden Möglichkeiten denkbar:

1. Der Standard wird von einer offiziellen Standardisierungsorganisation oder einem eigens gegründeten Konsortium entwickelt. Alle Firmen können sich an der Entwicklung beteiligen und den Standard nützen.
2. ModernCooking und ausgewählte Partner entwickeln den Standard. Andere Firmen können den Standard adaptieren.
3. ModernCooking entwickelt den Standard allein, erlaubt aber anderen Unternehmen die Nutzung.

Da bis jetzt alle Argumente für eine Compliance-Strategie sprechen, nehmen wir in den folgenden Überlegungen an, dass in jedem dieser Fälle zusätzlich eine Compliance-Marke eingeführt wird, die Firmen freiwillig erwerben können.

Die erste Möglichkeit erreicht die größte Akzeptanz. Die Wahrscheinlichkeit, dass Unternehmen einen Konkurrenzstandard starten, ist gering. Da sich alle Firmen am Standard beteiligen, wird der Mix&Match Nutzen der Konsumenten optimiert und ihnen die Angst vor Lock-In genommen.

Eine mögliche Gefahr ist, dass niemand die Rolle des sponsors übernimmt. Die Aufgabe eines sponsors wäre in diesem Fall Produkte in der Anfangsphase billiger abzugeben, vertikale Kooperationspartner zu engagieren und die Fragmentierung des Standards zu verhindern.

Allerdings können auch im zweiten und dritten Fall Konkurrenten nicht von dem Standard ausgeschlossen werden. Daher bieten auch diese beiden Formen nicht mehr Anreize für Unternehmen in den Standard zu investieren, es ist lediglich das Image einiger Unternehmen direkter mit dem Erfolg des Standards verknüpft und die Verantwortung für die Verbreitung des Standards klarer zugeordnet. Wenn zwei Unternehmen einen Standard einführen, dann ist eben auch klar, dass diese beiden Unternehmen den Standard sponsern.

Durch ein eigenes Konsortium, das für Entwicklung und Sponsorship des Standards verantwortlich ist, lässt sich diese Verantwortung bei einer für alle offenen Kooperation besser zuordnen, als durch eine lose Zusammenarbeit oder die Entwicklung durch eine unabhängige Organisation. Dieses Konsortium wird von all jenen Unternehmen finanziert, die den Standard nützen. Ob und wie so ein Konsortium trotz public-good Charakters des Standards aufgebaut werden kann, wird in der nächsten Phase diskutiert.

Durch diese offene Kooperation wird eine maximale Marktbelegung erreicht. Auch für Einnahmen aus dem Portal ist eine möglichst große Verbreitung des Standards vorteilhaft. Die anderen Finanzierungsmöglichkeiten sind bei dieser Kooperationsstrategie allerdings wenig Erfolg versprechend. Wenn der Standard von einem Konsortium aller Firmen eingeführt wird, bringt er ModernCooking weder Imagevorteile noch einen wesentlichen Anteil an Compliance-Einnahmen und gebundenen Kunden.

Bei einer Kooperation mit einer begrenzten Anzahl von Unternehmen müsste darauf geachtet werden, dass die Koalitionsteilnehmer gemeinsam über eine vollständige Produktpalette verfügen und zumindest ein Teil der großen Player beteiligt ist. In diesem Fall würde der Nutzen aus Mix&Match für Anwender nur geringfügig sinken und auch die Gefahr von excess inertia würde sich nicht allzu viel erhöhen.

Steigen würde allerdings die Wahrscheinlichkeit, dass ein anderes Unternehmen einen Konkurrenzstandard startet, da Eintrittsbarrieren gering sind und auch die Stärke der Netzwerkeffektivitäten zwei Standards zulässt.

Die beiden Nachteile der geringeren Verbreitung des Standards und der möglichen Konkurrenz durch einen zweiten Standard müssen nun gegen die Vorteile einer kleineren Koalition abgewogen werden. Diese Vorteile liegen darin, dass die wenigen einführenden Unternehmen den Standard vielleicht besser mit ihrem Firmennamen verknüpfen können, sie mehr aus Compliance-Tests einnehmen und mehr Kontrolle über den Standard haben. Die drei Vorteile scheinen aber zu wenig, um das Risiko eines Konkurrenzkampfes zwischen zwei Standards einzugehen.

Sowohl Vor- als auch Nachteile werden noch verschärft, wenn ModernCooking versucht den Standard alleine einzuführen. Denn bei einer alleinigen Einführung kann ModernCooking noch stärkere Imagevorteile verbuchen und hat alleinige Kontrolle über Entwicklung des Standards. Allerdings sinkt die zugesicherte Unterstützung noch weiter. Da ModernCooking zu Beginn das einzige Unternehmen ist, das dem Standard verpflichtet ist, werden Konsumenten zögern den Standard zu adaptieren, bevor klar ist, wie die großen Firmen reagieren. Denn zu Beginn verfügt der Standard nur über eine kleine Produktpalette und es bieten sich wenig Mix&Match Möglichkeiten. Auch die anderen Unternehmen werden zögern einen proprietären Standard zu adaptieren, solange dieser weder unimitierbare Funktionen noch ein großes Netzwerk bietet. In dieser Situation ist die Wahrscheinlichkeit hoch, dass der Standard an excess inertia scheitert.

Nach diesen Überlegungen scheint eine für alle offene Koalition die beste Lösung. Wie würde solch eine Kooperation den Wettbewerb verändern? Sie würde auf jeden Fall den Wettbewerb von den Systemen zu den Komponenten verlagern. Da alle Komponenten interagieren können, müssen sich Anwender nicht für das System von Siemens, Miele oder ModernCooking entscheiden, sondern können die Komponenten beliebig kombinieren. Das ist vor allem für kleine Unternehmen wie ModernCooking ein Vorteil, da ihre Systeme unvollständig sind und sie daher auf Komponentenebene besser konkurrieren können.

Hingegen ist nicht zu befürchten, dass eine große Kooperation Innovationen hemmt oder Unternehmen zu wenig Differenzierungsmöglichkeiten bietet. Alle Möglichkeiten zur Differenzierung, die Küchengeräte bisher geboten hatten, bleiben bestehen und durch neue Funktionen werden zusätzliche Möglichkeiten geschaffen, die durch einen gemeinsamen Rezeptstandard kaum eingegrenzt werden.

In der Frage der horizontalen Kooperation scheint also die Einrichtung eines Konsortiums sinnvoll, das für die Entwicklung und Verbreitung des Standards verantwortlich ist und dem jedes interessierte Unternehmen beitreten kann.

Es bietet sich an, auch vertikale Kooperationspartner an diesem Konsortium zu beteiligen. Diese Partner bringen einerseits ihr Know-How, etwa aus Rezeptdatenbanken oder Gastronomie-Management Systemen, in die Entwicklung des Standards ein und garantieren andererseits breitere Unterstützung.

Da es weder zielführend noch wahrscheinlich ist, dass sich alle Rezeptdatenbanken und andere vertikale Partner an dem Konsortium beteiligen, ist es ebenfalls Aufgabe des Konsortiums nicht beteiligte Unternehmen zur Verwendung des CookML Standards zu motivieren. Ein Beispiel wäre ein Programm, das Rezeptdatenbanken bei der Konvertierung bestehender Rezepte nach CookML unterstützt. Solche Maßnahmen werden in der nächsten Phase beim Verhindern von excess inertia diskutiert.

Nun bleibt noch zu klären, ob auch für das Portal eine Kooperation sinnvoll ist. Eigentlich müsste auch für die Einführungsstrategie des Portals das ganze Framework durchlaufen werden, doch wir nehmen das Ergebnis vorweg und setzen voraus, dass das Portal alleine eingeführt werden soll. Denn Ziel des Portals ist es ja das eigene Image zu stärken und direkte Erlöse zu erwirtschaften. Es besteht zwar die Gefahr von excess inertia und Konkurrenz, aber dieses

Risiko kann getragen werden, da ein Scheitern des Frameworks nur ein kleines Segment von ModernCookings Einnahmen gefährdet. Bei einer Kooperationsstrategie würden diese Einnahmen in jedem Fall geringer ausfallen.

Daher muss ModernCooking versuchen das Konsortium so (mit)zugestalten, dass es nicht selbst ein solches Portal aufbaut. Weiters wäre es für den Aufbau des Portals vorteilhaft über Funktionen zu verfügen, die nur von ModernCooking angeboten werden. Einige solcher Detailfunktionen werden in der nächsten Phase beschrieben.

Offenheit

Viele Elemente der Offenheitsstrategie wurden durch die Natur des Rezeptstandards und die Entscheidung für die Implementierung eines offenen Konsortiums bereits vorweggenommen. Eine Kooperation, der jeder beitreten kann, und ein Standard, den jeder nützen kann, verlangen eine vollkommen offene Implementation und Entwicklung. Für Unternehmen bieten sich in diesem Modell keinerlei Wettbewerbsvorteile, wenn der Zugang in irgendeiner Form eingeschränkt wird. Betrachten wir die Skalen aus Kapitel 7.1 so bleiben nur mehr zwei Argumente die Offenheit des Standards einzuschränken: Die Gefahr der Fragmentierung und die des fehlenden Sponsors.

Die Implementation des Standards soll jedem offen stehen. Wenn es keinen Grund gibt die Offenheit einzuschränken, dann sollte jedermann die Verwendung des Standards so leicht wie möglich gemacht werden, um die Netzwerkexternalitäten zu maximieren. Das heißt die Spezifikation des Standards muss aktuell, einfach zugänglich und ausführlich dokumentiert zur Verfügung stehen. Beispielrezepte, Tutorials und Tools zum korrekten Erstellen von Rezepten erleichtern die Implementation.

Lediglich Compliance-Tests schränken die Freiheit der Implementation ein, wenn auch aufgrund ihrer Freiwilligkeit nur geringfügig. Die Einnahmen aus der Compliance-Zertifizierung können als Wettbewerbsvorteil des Konsortiums gegenüber nicht beteiligten Unternehmen gesehen werden. Dieser Ansatz kann verwendet werden, um Unternehmen zu motivieren sich an der Koalition zu beteiligen.

Um eine offene Entwicklung zu gewährleisten muss ein Weg implementiert werden, wie Unternehmen und Anwender Änderungswünsche am Standard deponieren können. Die Entscheidung, welche Änderungen in die nächste Spezifikation aufgenommen werden, liegt allerdings bei dem Konsortium. Da nur die vom Konsortium veröffentlichte Version gültig ist, ist durch diese Vorgehensweise keine Gefahr einer „erlaubten“ Fragmentierung gegeben. Es besteht allerdings die Gefahr, dass Unternehmen den Standard unvollständig oder nicht ganz korrekt in ihren Produkten implementieren. Dieses Risiko soll durch die Compliance-Prüfung gesenkt werden.

Eine schwierige Frage ist die Anpassbarkeit des Standards. Denn einerseits ist ein hohes Maß an Flexibilität gefordert, da Kochvorgänge und Zutaten höchst heterogen sind. Doch andererseits reduziert Flexibilität den Grad der Automatisierung und kann zur Fragmentierung führen.

Freie Bereiche sollten daher so sparsam wie möglich verwendet werden und es gilt zu verhindern, dass Anwender das gesamte Rezept in die freien Bereiche schreiben. Auch dafür ist ein gut dokumentiertes Gratis-Tool zur Erstellung von Rezepten hilfreich.

Ein möglicher Mittelweg der Anpassbarkeit soll am Beispiel von Kochvorgängen erklärt werden: Eine große Anzahl von Kochvorgängen, wie Braten, Grillen oder Kochen, wird vordefiniert und muss durch einen eindeutigen Code identifiziert werden können. Wenn ein Rezept nun einen nicht definierten Kochvorgang enthält, dann muss der Anwender dafür ein eigenes Element verwenden, in dem er eine verbale Beschreibung des Vorgangs angibt. Diese nicht definierten Elemente *dürfen* aber von Küchengeräten nur verbal wiedergegeben und nicht automatisch verarbeitet werden. Dadurch erreicht man, dass Küchengerätehersteller nicht beginnen ihren eigenen

proprietären Kochvorgänge zu definieren und in ihren Geräten zu interpretieren. Produzenten können aber beim Konsortium die Definition neuer Kochvorgänge beantragen, sodass sie in der nächsten CookML-Version berücksichtigt werden.

Weiters darf der Standard nur in der Spezifikation definierte Elemente enthalten. Es ist nicht, wie zum Beispiel in HTML, erlaubt zusätzliche Elemente zu erfinden, die von manchen Programmen interpretiert und von anderen ignoriert werden. Daher müssen freie Bereiche zur Verfügung stehen, in denen verbale Beschreibungen für Vorgänge abgelegt werden können, die sich wirklich mit keinem vorhandenen Element abbilden lassen. Denn fehlen solch freie Bereiche gänzlich, so sind Anwender gezwungen von der CookML-Spezifikation abzuweichen, wenn sie mit den vorgeschriebenen Elementen, Rezepte nicht korrekt abbilden können.

Damit ist die zweite Phase abgeschlossen und die drei Grundelemente der Strategie sind definiert.

10.3 Phase 3 - Feinabstimmung der Strategie

Im letzten Schritt werden nun weitere Details der Einführungsstrategie ausgearbeitet. Immer wieder wird die Frage auftreten, welche Maßnahmen von den einzelnen Firmen und welche vom Konsortium gesetzt werden sollen. Für ModernCooking inkludiert das auch die Überlegung, welche Funktionen nur auf ihrem proprietären Portal zur Verfügung gestellt werden sollen, um sich von Konkurrenten zu differenzieren.

Verhindern von Fragmentierung

Wie bei der Offenheitsstrategie besprochen wurde, besteht durchaus die Gefahr einer Fragmentierung. Die bisherigen Lösungsansätze sind die Compliance-Prüfung, kontrollierte freie Bereiche der Rezepte und ein Tool zur korrekten Erstellung von Rezepten.

Die Compliance-Prüfung ist umso effektiver, je bekannter die Compliance-Marke ist und je mehr Konsumenten darauf achten nur zertifizierte Produkte zu verwenden. Neben dieser Compliance-Prüfung für professionelle Anbieter von Software, Küchengeräten und Rezepten könnte noch eine Low-End Compliance Zertifizierung für private Anwender eingeführt werden. Diese Zertifizierung könnte über eine kostenlose, automatische Prüfung auf einer Website erfolgen, die einerseits die korrekte Syntax von Rezepten kontrolliert und andererseits versucht Tipps zur besseren Formulierung des Rezeptes zu geben. Über diese Prüfung ist es nur möglich einzelne Rezepte zu prüfen und sie berechtigt nicht zum Führen der Compliance-Marke, sondern fügt nur dem Rezept eine Bestätigung hinzu, dass das Rezept geprüft wurde. Rezepttauschbörsen könnten eine solche Bestätigung als Bedingung voraussetzen, um Rezepte zum Tausch zu akzeptieren. Denkbar ist auch, dass Rezepttauschbörsen diese Überprüfung selbst durchführen können.

Das Tool zur Erstellung der Rezepte sollte genauso wie die Compliance-Prüfungen von dem Konsortium angeboten werden. Denn diese beiden Funktionen können die Verbreitung des Standards entscheidend beeinflussen und sollten daher so öffentlich wie möglich zur Verfügung stehen. Der Erfolg des proprietären Portals ist noch ungewiss und daher sollten so wichtige Funktionen nicht nur dort verfügbar sein.

Vorankündigung und Konsumentenerwartungen

Wenn es gelingt, die großen Unternehmen des Marktes für die Koalition zu gewinnen, stellen negative Konsumentenerwartungen kein Problem dar. Vorankündigungen werden vor allem dann eingesetzt, wenn zwei konkurrierende Standards um die Vorherrschaft kämpfen und sind daher

in diesem Fall auch nicht nötig. Wichtig ist allerdings, dass das Vertrauen der Konsumenten in den Standard nicht bereits in der Einführungsphase enttäuscht wird, da noch zu wenige Rezepte zur Verfügung stehen um den Standard sinnvoll zu nutzen. (siehe nächster Punkt)

Ausmaß der Netzwerkexternalitäten managen

Zu Beginn ist es trotz breit angelegter Koalition wahrscheinlich, dass die Anzahl der verfügbaren Rezepte beschränkt ist. Vor allem der Austausch von Rezepten zwischen Anwendern kann erst nach einiger Zeit funktionieren, wenn genügend Anwender ihre Rezepte in CookML speichern. Daher ist es sinnvoll zu Beginn die Bedeutung der direkten Netzwerkexternalitäten zu reduzieren. Das heißt, der Austausch von Rezepten mit anderen Konsumenten wird zu Beginn nur wenig promotet, um Anwender, die versuchen Rezepte zu tauschen und nur wenig Angebot finden, nicht zu enttäuschen. Stattdessen könnten Anwender beim Kauf der Küchengeräte zum Beispiel ein kostenloses Jahresabo einer Kochzeitschrift geschenkt bekommen, das ihnen Zugang zu der CookML-Datenbank der Zeitschrift ermöglicht.

Diese Maßnahme ist wirkungsvoller, wenn sie von allen Anbietern umgesetzt wird, und könnte daher vom Konsortium organisiert werden. Verschenkt nur ModernCooking Zugang zu dieser Rezeptdatenbank, so dient die Aktion eher als Marketingmaßnahme, als um das allgemeine Vertrauen in den Standard zu heben.

Eine anfängliche Einführung in einem kleineren Netzwerk macht für den CookML-Standard keinen Sinn. Rezepte sind zumindest überregional, wenn nicht gar weltweit verwendbar. Ein Anwender in den USA bringt einem österreichischen Anwender genauso viel Nutzen wie ein weiterer Österreicher. Vereinfachung des weltweiten Austausches von Rezepten ist sogar ein wichtiger Vorteil des Rezeptstandards. Küchengeräte können fremde Maßeinheiten automatisch umrechnen und einheitliche Codes für Lebensmittel und Arbeitsschritte überwinden das Sprachproblem. Daher bietet eine möglichst internationale Einführung des Standards einen Mehrwert für Anwender.

Maßnahmen gegen excess inertia

Wenn es gelingt die geplante Koalition mit Unterstützung großer Partner aufzubauen, ist ein ausreichendes Angebot an CookML tauglichen Küchengeräten gesichert. Excess inertia droht dann vor allem dadurch, dass niemand Rezepte in CookML anbietet, bevor genügend Anwender den Standard nutzen und Anwender den Standard nicht verwenden, solange das Angebot an Rezepten klein ist.

Die Maßnahmen, die wir gegen diese Gefahr entwickeln, setzen an drei verschiedenen Punkten an. Sie erhöhen den stand-alone value für Anwender, den stand-alone value für Anbieter von Rezepten und beschleunigen das Wachstum des Netzwerks.

Ersteres lässt sich einmal dadurch erreichen, dass Anwender zu Beginn mit einer ausreichenden Anzahl von Rezepten versorgt werden. Dafür eignet sich das oben beschriebene Gratis-Abo oder eine mit Küchengeräten mitgelieferte CD mit einem Grundstock an Rezepten.

Mehrfach erwähnt wurde bereits ein kostenloses Tool zur einfachen Erstellung von CookML-Rezepten. In diesem Kontext bringt das einerseits einen höheren stand-alone value für Anwender. Sie können leicht ihre eigenen Rezepte nach CookML konvertieren und dadurch zum Beispiel Kochvorgänge automatisieren oder Kindern eine einfache Anleitung zum Kochen zur Verfügung stellen. Andererseits vergrößert sich dadurch auch das Netzwerk von CookML-Rezepten, da das Tauschen von Rezepten für Anwender weniger Aufwand darstellt. Eine weitere Möglichkeit

die Eingabe zu vereinfachen wäre das Rezept von den Küchengeräten während des Kochens automatisch aufzeichnen zu lassen.

Das Wachstum des Netzwerks lässt sich erhöhen, wenn auch Anwender ohne CookML-Küchengeräte den Standard nützen können. Das Konsortium könnte zum Beispiel CSS-Stylesheets zur Verfügung stellen, mit denen die Rezepte in jedem Webbrowser angezeigt werden können. Eine andere Möglichkeit ist ein Tool zur Verwaltung von privaten Rezeptsammlungen im CookML-Format, das zum Beispiel im Vergleich zu einer rein verbalen Speicherung von Rezepten eine detailliertere Suche ermöglicht.

Von der anderen Seite wird das Netzwerkwachstum erhöht, indem man versucht möglichst viele Rezeptdatenbanken für CookML zu gewinnen. Einerseits werden solche Anbieter durch direkte Kooperationen angeworben. So könnten zum Beispiel Gerätehersteller jenen Kochzeitschriften Abonnements für ihre Promotions abkaufen, die ihre Online-Rezeptdatenbanken auf CookML umstellen. Andererseits gilt es durch den Aufbau des Standards möglichst viele Anbieter von Rezepten anzusprechen. Wenn der Standard zum Beispiel ein Feld enthält, für welche Diäten sich dieses Rezept eignet, ist das ein großer Vorteil für jene, die Diäten entwickeln und vermarkten. Denn ernährungsbewusste Anwender können dann das Internet nach Rezepten durchsuchen, die für ihre Diät geeignet sind. Anbieter von Diäten erhalten so einen Anreiz ihre eigenen Rezepte ebenfalls auf CookML umzustellen.

Um den stand-alone value für Rezeptdatenbanken zu erhöhen könnte ModernCooking ein Programm zur Erstellung und Verwaltung von professionellen CookML-Datenbanken zur Verfügung stellen. Dadurch vereinfacht sich die interne Verwaltung für Rezeptdatenbanken und Such- und Auswertungsverfahren werden verbessert. Anbietern von Rezeptdatenbanken bietet der Umstieg auf den CookML-Standard damit einen internen Nutzen, unabhängig davon, ob Anwender die Rezepte nun in CookML oder einfach im Textformat herunterladen. Nebenbei könnte so ein Tool auch eine Schnittstelle für Meta-Suchmaschinen haben, die mehrere Rezeptdatenbanken durchsuchen. ModernCooking könnte dann auf seinem Portal eine solche Meta-Suchmaschine anbieten. Durch das hohe Potential dieser Verknüpfung ist es für ModernCooking sinnvoll ein solches Datenbank-Tool selbst zur Verfügung zu stellen und nicht dem Konsortium zu übergeben.

Sponsorship

Aufgrund der fehlenden Möglichkeit Konkurrenten auszuschließen und da Lock-In hauptsächlich in den Standard und nicht in die Produkte eines Unternehmens erfolgt, wird sponsorship relativ gering ausfallen.

CookML-Geräte werden zu Beginn zwar sicher verbilligt angeboten, da Konsumenten nicht bereit sind einen hohen Preis zu bezahlen, bevor ein ausreichendes Netzwerk aufgebaut wurde. Die Ermäßigungen werden allerdings geringer ausfallen, als wenn Konkurrenten von der installed base ausgeschlossen werden könnten.

Da die größte Gefahr von excess inertia beim Rezeptangebot liegt, sollten vor allem Rezeptdatenbanken gesponsert werden. Das geschieht einerseits durch Hilfe bei der Konvertierung der bestehenden Datenbank, zum Beispiel durch kostenlose Programme oder Beratung. Andererseits können zum Beispiel Compliance-Zertifikate im ersten halben Jahr gratis angeboten werden. So werden Rezeptdatenbanken motiviert schnell den neuen Standard zu adaptieren und sich zertifizieren zu lassen. Ein solches Angebot kann auch auf Zertifikate für andere Unternehmen ausgedehnt werden, um die Bekanntheit und Bedeutung der Compliance-Marke zu fördern.

Das Konsortium hilft hier free-ridership zu verhindern. Während jedes einzelne Unternehmen hofft auf der sponsorship der Konkurrenz free-riden zu können, erzwingt die vorherige Übereinkunft sich am Konsortium zu beteiligen Investitionen, von denen alle profitieren.

Ihr eigenes Portal werden ModernCooking allerdings intensiv durch sponsorship unterstützen, da nur sie davon profitieren und eine höhere Gefahr von excess inertia besteht. So wird zum Beispiel das oben erwähnte Tool für Rezeptdatenbanken oder die Meta-Suchmaschine kostenlos angeboten.

Managen der Switching Costs

Switching costs beim Wechsel zum Rezeptstandard bestehen vor allem für Rezeptdatenbanken und werden, wie schon beschrieben, durch Konvertierungshilfen gesenkt.

Wenn CookML, wie geplant, der einzige Rezeptstandard ist, treten switching costs vor allem beim Wechsel zu billigeren Geräten, die keine Rezepte verarbeiten können auf. Der Lock-In besteht darin, dass Anwender wenn sie einmal ein CookML-Gerät erworben haben, auch in Zukunft teurere CookML-Geräte kaufen.

Die Hilfen für Anwender zur einfachen Erstellung von Rezepten bringen Anwender auch dazu ihre eigenen switching costs zu erhöhen, da sie leichter Informationsbestände aufbauen können.

Die switching costs durch langlebige Güter werden erhöht, indem Anwendern möglichst viele komplementäre Geräte angeboten werden. Da die Geräte nicht alle gleichzeitig ersetzt werden, bietet sich dann kein geeigneter Zeitpunkt für einen Wechsel des Standards. Diese switching costs sind allerdings unwichtig, solange es keinen Konkurrenzstandard gibt, da Anwender den Standard nicht wechseln sondern nur verlassen können und das Verlassen auch schrittweise möglich ist. Die aufgebauten Informationsbestände und das angeeignete Wissen über die Technologie verlieren hingegen nicht an Wert. Anwender sind durch sie an den Standard gebunden, da sie die getätigten Investitionen nicht verlieren wollen.

Als Unterscheidungskriterium von gebundenen und neuen Usern eignet sich der Kauf einer Zentraleinheit. Denn Anwender, die ihr erstes CookML-Gerät kaufen, müssen gleichzeitig auch eine Zentraleinheit erwerben. Daher können Zentraleinheiten *in Kombination* mit einem Gerät billig abgegeben werden, um User erstmalig an den Standard zu binden. Gebundene User, die ihre Zentraleinheit ersetzen möchten, kaufen diese meist nicht gleichzeitig mit einem Küchengerät und können daher von neuen Usern unterschieden werden. Da von den gebundenen Usern aber alle Unternehmen gleichermaßen profitieren, werden Preisermäßigungen auch hier geringer ausfallen.

Erfolg versprechender ist für ModernCooking in den Küchengeräten außerhalb des CookML-Standards proprietäre Elemente zu verwenden. So könnten die Geräte Rezept-Favoriten verwalten, Vorlieben der Anwender intelligent erlernen und weitere Konfigurationen speichern, die beim Austausch von Geräten nur weiterverwendet werden können, wenn auch das Nachfolgegerät von ModernCooking ist. Dadurch werden switching costs innerhalb des Standards aufgebaut, die Anwender nicht nur an den Standard sondern direkt an ModernCooking binden.

Zeitplan

Da der first-mover advantage gering ist und es voraussichtlich nur einen Rezeptstandard geben wird, hat das Konsortium keinen großen Zeitdruck bei der Einführung. Es ist wichtiger zum Zeitpunkt der Einführung eine breite Unterstützung auf allen Ebenen aufgebaut zu haben, als den Standard möglichst rasch einzuführen.

Für ModernCooking selbst und seine Bemühungen sich innerhalb der Koalition zu profilieren ist der Zeitplan aber sehr wohl wichtig. Als Initiator der Koalition hat ModernCooking am

längsten Zeit CookML-Produkte zu entwickeln und sollte daher versuchen die Produkte als Erster auf den Markt zu bringen. Dadurch sichern sie sich aber vor allem Publicity, da der first-mover advantage für Küchengeräte nur schwach ist.

Bei der Einführung des Portals spielt der first-mover advantage allerdings eine entscheidende Rolle. Durch das Datenbank-Tool für Rezeptanbieter kann ModernCooking eine Verbreitung von CookML und seinem Portal unabhängig von der langsamen Adaption von Küchengeräten erreichen. Wenn sie mit ihrem Portal als erster am Markt sind, besteht eine realistische Chance damit einen hohen Bekanntheitsgrad zu erreichen.

Free-ridership nützen oder verhindern

Da der Standard keinen Ausschluss von Konkurrenten ermöglicht, wurde er schon sehr offen gestaltet und erlaubt daher prinzipiell free-ridership. Die einzige Gefahr der free-ridership ist, dass sich große Unternehmen nicht an der Koalition beteiligen, da sie den Standard auch ohne Beteiligung genauso gut nützen können. Diese Probleme beim Aufbau einer Koalition werden im nächsten Schritt besprochen.

Solange sich genügend Unternehmen an der Kooperation beteiligen, um positive Konsumentenerwartungen zu erreichen und eine ausreichende Produktpalette anzubieten, ist in dieser offenen Strategie jede Adaption des Standards von free-ridenden Unternehmen erwünscht, um das Netzwerk zu vergrößern.

Kooperation aufbauen

Dieser Faktor ist der kritische Punkt der Strategie. Die Strategie scheint die Verbreitung des Standards zu sichern, für ModernCooking maximale Gewinne und für Anwender hohen Nutzen zu bringen. Doch alles ist davon abhängig, dass sich eine Koalition mit Beteiligung der großen Produzenten von Küchengeräten bilden lässt.

Es ist durchaus möglich, dass große Unternehmen es nicht für nötig halten sich an der Entwicklung des Standards zu beteiligen. Wenn der Standard sich durchsetzen können sie ihn immer noch adaptieren, andernfalls führen sie einen eigenen proprietären Rezeptstandard ein.

ModernCooking hat im wesentlichen drei Argumente, um Unternehmen zur Kooperation zu motivieren. Erstens soll das Konsortium durch Einnahmen aus den Compliance-Prüfungen finanziert werden. Die Kosten, die einem Unternehmen durch eine Kooperation entstehen, sind also gering. Zweitens lässt sich der Standard nur mit Hilfe der großen Unternehmen durchsetzen. ModernCooking muss den potentiellen Partnern klarmachen, welche Entwicklungsmöglichkeiten ein gemeinsamer Standard für den Markt bietet und welche Vorteile der gemeinsame Standard gegenüber proprietären Standards einzelner großer Unternehmen hat. Weiters ist es unsicher, ob es für große Unternehmen möglich ist abzuwarten und den Standard später zu adaptieren, da sich der Standard ohne Unterstützung der großen Unternehmen unter Umständen nie durchsetzen kann. Drittens kann das Unternehmen im Konsortium die Entwicklung des Standards direkt beeinflussen.

Zumindest zu Beginn sollte es nicht möglich sein dem Konsortium nachträglich beizutreten. Dadurch kann verhindert werden, dass alle Unternehmen abwarten, wie sich der Standard entwickelt. Wenn ein Unternehmen sich geweigert hat an der Kooperation teilzunehmen, hat es keine Chance mehr die Entwicklung des Standards mitzubestimmen und von Einnahmen aus Compliance-Prüfungen zu profitieren.

Je mehr größere Unternehmen sich der Kooperation angeschlossen haben, umso leichter wird es weitere zu verpflichten. Der schwierigste Schritt wird sein die ersten Partner zu gewinnen.

10.4 Zusammenfassung

Die entscheidenden Punkte der vorgeschlagenen Strategie sollen hier noch einmal kurz zusammengefasst werden:

- Vorrangiges Ziel ist die Belebung des Marktes.
- Die wichtigsten Unternehmen, die CookML-Produkte anbieten wollen, gründen gemeinsam ein Konsortium, das den Standard entwickelt und sponsert.
- Jeder, egal ob Mitglied der Koalition oder nicht, darf CookML kostenlos benutzen.
- Unternehmen können ihre Produkte auf korrekte Implementation der CookML-Spezifikation testen lassen und dürfen dafür eine Compliance-Marke führen.
- Zur Vermeidung von excess inertia werden zahlreiche Detailmaßnahmen getroffen. Eine der wichtigsten ist ein kostenloses Tool zur komfortablen Erstellung von CookML Rezepten.
- Der kritische Erfolgsfaktor ist die Frage, ob es gelingt die großen Unternehmen zur Kooperation zu bewegen.
- ModernCooking wird versuchen sich innerhalb der Koalition mit Hilfe eines umfassenden, proprietären Portals für CookML-Services zu profilieren.

Bei der Anwendung des Frameworks hat sich gezeigt, wie flexibel das Framework gehandhabt werden muss. So haben sich zum Beispiel in dieser Situation zwei stand-alone values, einer für die Küchengeräte und einer für den Rezeptstandard ergeben, die beide in die Überlegungen miteinbezogen wurden. Die Offenheitsstrategie wurde größtenteils durch die Kooperationsstrategie vorherbestimmt und diente nur noch zum Klären von Details. Mit dem Portal ist aus dem Framework eine neue Geschäftsidee entstanden, für die mit einer neuerlichen Anwendung des Frameworks wieder eine Strategie ausgearbeitet werden muss. Wenn der Aufbau der Koalition scheitern sollte, wird ModernCooking einen Schritt zurückgehen müssen und das Framework mit einer geänderten Kooperationsstrategie nochmals durchlaufen. Alle diese Überlegungen sind im Framework nicht direkt abgebildet, müssen aber in die Entwicklung einer Strategie miteinbezogen werden. Das Framework systematisiert und erleichtert die Entwicklung, erspart aber nicht die detaillierte Analyse der Situation.

Wie Produzenten von Haushaltsgeräten mit Standardisierung in der Realität umgehen, hat sich an der Entwicklung eines Standards zur Kommunikation zwischen Haushaltsgeräten gezeigt. Dieser Standard soll unterschiedlichen Geräten ermöglichen über Stromleitungen oder Funk Informationen auszutauschen. Die Entwicklung des Standards wird im Rahmen der CECED, dem Verband europäischer Haushaltsgerätehersteller durchgeführt und soll bis Ende 2002 abgeschlossen sein[35]. Als Voraussetzung bei der Entwicklung werden kostenloser Zugang zur Spezifikation, keine Lizenzgebühren und keine Patentierung des Standards angeführt, es soll also ein sehr offener Standard eingeführt werden. Gründe für einen einheitlichen Standard liegen in der gemeinsamen Nutzung der Zentraleinheit, geteilten Entwicklungskosten, Konsumentenvertrauen und den Möglichkeiten der Zertifizierung[52]. Die reale Strategie für die Entwicklung dieses Standards ist also dem Ergebnis unseres Frameworks für den fiktiven Rezeptstandard durchaus ähnlich.

Literaturverzeichnis

- [1] ADOBE SYSTEMS INC.(1999), Portable Document Format Reference Manual, <http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspect.pdf>
- [2] AUSTRIAN SMART CARD ASSOCIATION(2000), Quick-Wertkarten und Affinity-Wertkarten, ASA News, Juli, <http://www.asa.or.at/asanews200007.pdf>
- [3] BERLIND, D.(2002), IBM pressures Sun to free Java, Zdnet, 11. September, <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2879892-1,00.html>
- [4] BESEN, S.M., FARRELL, J.(1994), Choosing How to Compete: Strategies and Tactics in Standardization, Journal of Economic Perspectives, Vol.8, No. 2
- [5] BODOW, S.(2001), The Money Shot, Wired 9.09, September, http://www.wired.com/wired/archive/9.09/paypal_pr.html
- [6] BROERSMA, M.(2002), Tech doesn't buoy Netscape browser, News.com, 28.August, <http://news.com.com/2100-1023-955734.html>
- [7] CARGILL, C.(1997), Sun and the Standardization Wars, StandardView 5(4)
- [8] COCCA, T. D., CSOPORT, P.(2001), Standortbestimmung zum elektronischen Geld, Der Schweizer Treuhänder 12/01
- [9] DAVID, P.A.(1995), Standardization policies for network technologies: the flux between freedom and order revisited, in: Standards, Innovation and Competitiveness, Hrsg.: Hawkins et al., Edward Elgar, Aldershot und Brookfield
- [10] DE VRIES, H. J., HENDRIKSE, G. W. J.(2001), The Dutch banking Chipcard Game: Understanding a Battle Between Two Standards, ERIM Report Series in Management, No. ERS-2001-18-ORG, <http://www.eur.nl/WebDOC/doc/erim/erimrs20010322174833.pdf>
- [11] ECONOMIDES, N.(1996), The Economics of Networks, International Journal of Industrial Organization, <http://www.stern.nyu.edu/networks/94-24.pdf>
- [12] FARRELL, J., SALONER, G.(1985), Standardization, Compatibility and Innovation, Rand Journal of Economics 16, No.3
- [13] FARRELL, J., SALONER, G.(1986), Installed Base and Compatibility: Innovation, Product Preannouncements and Predation, American Economic Review 76, No. 5
- [14] FARRELL, J.(1995), Arguments for weaker Intellectual Property Protection in Network Industries, StandardView, Vol.3, No.2, June

- [15] FESTA, P.(2002), Mozilla finally turns 1.0, News.com, 5.Juni,
<http://news.com.com/2100-1023-932814.html>
- [16] FORAY, D.(1995), Coalitions and committees: How users get involved in information technology standardization, in: Standards, Innovation and Competitiveness, Hrsg.: Hawkins et al., Edward Elgar, Aldershot und Brookfield
- [17] FORMATDATA(2002), RecipeML - The Format for Online Recipe Content,
<http://www.formatdata.com/recipe/ml/>
- [18] GILLIES, C.(2002), Internet-Bezahldienst Paypal konkurriert mit Großbanken, WebWelt, 23.Juli, <http://www.welt.de/daten/2002/07/23/0723wa346133.htm>
- [19] GLANZ, A.(1992), Ökonomie von Standards, Europäische Hochschulschriften, Peter Lang, Frankfurt a. M.
- [20] GLEICK, J.(2000), Patently Absurd, New York Times, March 12,
<http://www.nytimes.com/library/magazine/home/20000312mag-patents.html>
- [21] HAGENLOCHER, C., SELLS, C.(2002), A Roadmap to the Recently Released Windows APIs, O'Reilly, 30 August,
<http://www.oreillynet.com/pub/a/dotnet/2002/08/28/winapi.html>
- [22] HAWKINS, R.,MANSELL, R., SKEA, J.(1995), Standards, Innovation and Competitiveness, Edward Elgar, Aldershot and Brookfield
- [23] HEISE(2000), Real und Microsoft sträuben sich gegen Streaming-Standard, Heise.de, 18. Dezember, <http://www.heise.de/newsticker/data/nij-18.12.00-001/>
- [24] JAKOBS, K.(2000), Standardisation Processes in IT, Viehweg, Braunschweig
- [25] JAVAWORLD(1997), IBM joins Java apostles in 40-city tour to woo developers, JavaWorld, 24. Februar,
<http://www.javaworld.com/javaworld/jw-03-1997/jw-03-infoworld.worldtour.html>
- [26] KATZ, M., SHAPIRO, C.(1985), Network Externalities, Competition and compatibility, American Economic Review 75, No. 3
- [27] KATZ, M., SHAPIRO, C.(1986), Technology Adoption in the Presence of Network Externalities, Journal of Political Economy 94, No. 4
- [28] KINDLEBERGER, C.P.(1983), Standards as Public, Collective and Private Goods, KYKLOS, Vol. 36, S. 377- 396
- [29] KLEMPERER, P.(1995), Competition when Consumers have Switching Costs: An Overview with Applications to Industrial Organization, Macroeconomics, and International Trade, <http://www.nuff.ox.ac.uk/users/klemperer/competition.pdf>
- [30] KOBAK, J.B.(1998), Intellectual Property, Competition Law and Hidden Choices Between Original and Sequential Innovation, Virginia Journal of Law and Technology
- [31] LIBICKI, M., SCHNEIDER, J., FRELINGER, D. R., SLOMOVIC, A.(2000), Scaffolding the New Web:Standards and Standards Policy for the Digital Economy,
<http://www.rand.org/publications/MR/MR1215/>

- [32] MARKOFF, J.(1999), Toiling Not So Secretly in Switzerland, a Windows Emulator Gains on Microsoft, New York Times, January 18,
<http://www.nytimes.com/library/tech/99/01/biztech/articles/18linux-microsoft.html>
- [33] MATUTES, C., REGIBEAU, P.(1988), Mix and Match: Product Compatibility Without Network Externalities, Rand Journal of Economics 16, No.3
- [34] MERTZ, M.(1997), The Introduction of New Technology in the Presence of Network Externalities, Diplomarbeit, Universität Wien
- [35] MIELE(2002), Miele@home 2002, <http://www.miele.at>
- [36] MOZILLA(2000), The Mozilla Project and mozilla.org, Juni,
<http://www.mozilla.org/mozorg.html>
- [37] NETSCAPE(1998), NETSCAPE ANNOUNCES PLANS TO MAKE NEXT-GENERATION COMMUNICATOR SOURCE CODE AVAILABLE FREE ON THE NET, 22. Jänner, <http://wp.netscape.com/newsref/pr/newsrelease558.html>
- [38] NETSCAPE(1998), NETSCAPE ANNOUNCES MOZILLA.ORG, A DEDICATED TEAM AND WEB SITE SUPPORTING DEVELOPMENT OF FREE CLIENT SOURCE CODE, 23. Februar, <http://wp.netscape.com/newsref/pr/newsrelease577.html>
- [39] NETSCAPE(2000), Netscape Gecko Overview, <http://wp.netscape.com/browsers/gecko/>
- [40] NUVANTAGE(2001), The Paypal Phenomenon,
http://www.nuvantage.com/pdfs/NuVantage_PayPal_Phenomenon.pdf
- [41] PFLÜGER, J., PURGATHOFER, P.(2002), FAQ: Microsoft
- [42] RICCIUTI, M.(1998), Platform ploys in the public eye, News.com, 2. Dezember,
<http://news.com.com/2009-1001-218437.html>
- [43] ROHLFS, J.(1974), A Theory of Inderdependent Demand for a Communications Service, Bell Journal of Economics 5, No.1
- [44] SCHULMAN, A., MAXEY, D., PIETREK, M.(1994), Undocumented Windows: A Programmers Guide to Reserved Microsoft Windows Api Functions, Addison-Wesley, Reading, Massachusetts
- [45] SHANKLAND, S., KANELLOS, M., WONG, W.(2001), Sun, Microsoft settle Java suit, News.com, 23. Jänner, <http://news.com.com/2100-1001-251401.html>
- [46] SHANKLAND, S.(2002), Sun looks for payoff to Java addiction, ZDNet.com, March 26,
<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,5105279,00.html>
- [47] SHAPIRO, C., VARIAN, H.(1998), Locked In, Not Locked Out, The Industry Standard, 23. Oktober, <http://www.thestandard.com/article/display/0,1151,2173,00.html>
- [48] SHAPIRO, C., VARIAN, H.(1999), Information Rules: A Strategic Guide to the Network Economy, Harvard Business School Press, Boston, Massachusetts
- [49] SLIWA, C.(1998), Sun bolsters process for Java standards, Computerworld, 12. Oktober,
<http://www.computerworld.com/news/1998/story/0,11280,32980,00.html>

- [50] SUN(1997), Sun sues Microsoft for Breach of Java Contract,
<http://java.sun.com/pr/1997/oct/pr971007.html>
- [51] SUN(2002), Lizenz für Java 2 SDK, Standard Edition 1.4.1., <http://java.sun.com>
- [52] TRIALOG(2002), The Need of a European Standard for Power-line Communciation in Home Appliances, <http://www.trialog.com/Pdf/Euroforum2002Trialog.pdf>
- [53] WARREN-BOULTON, F.R., BASEMAN, K.C., WORROCH, G.(1996), Economics of intellectual property protection for software: the proper role for copyright, StandardView, Vol.3, No.2, June
- [54] WEISS, M., SPRING, M.(1992), Selected Intellectual Property Issues in Standardization, <http://www2.sis.pitt.edu/spring/papers/stdip1.pdf>
- [55] WEISS, M.B.H., TOYOFUKU, R. T.(1996), Free-Ridership in the Standards-Setting Process: The Case of 10BaseT, <http://www2.sis.pitt.edu/mweiss/papers/free-rider.pdf>
- [56] WINE(2002), Wine Development HQ, <http://www.winehq.com/>
- [57] WOLVERTON, T.(2002), PayPal hit by class-action suit, News.com, 21. Februar, <http://news.com.com/2100-1017-842240.html>
- [58] WOLVERTON, T.(2002), Sotheby's, eBay to launch joint site, ZDnet, 10. Juni, <http://zdnet.com.com/2100-1106-934884.html>
- [59] ZDNET(2000), PayPal no friend to online buyers, ZDNet, 19. Juli, <http://zdnet.com.com/2100-11-502663.html?legacy=zdn>