HOW OPEN SHOULD AN OPEN SYSTEM BE?

Essays on Mobile Computing

by

Kevin J. Boudreau

B.A.Sc., University of Waterloo
M.A. Economics, University of Toronto

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of Doctor of Philosophy
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

© 2006 Massachusetts Institute of Technology. All Rights Reserved.

The author hereby grants to Massachusetts Institute of Technology permission to reproduce and to distribute copies of this thesis document in whole or in part.

Signature of Author

Sloan School of Management

3 May 2006

Certified by

Rebecca Henderson
Eastman Kodak LFM Professor of Management
Thesis Supervisor

Michael Cusumano
Sloan Management Review Professor of Management
Thesis Supervisor

Marc Rysman
Assistant Professor of Economics, Boston University
Thesis Supervisor

Accepted by

Birger Wernerfelt
J. C. Penney Professor of Management Science and Chair of PhD Committee
HOW OPEN SHOULD AN OPEN SYSTEM BE?

Essays on Mobile Computing

by

Kevin J. Boudreau

Submitted to the Sloan School of Management
on 3 May 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

“Systems” goods—such as computers, telecom networks, and automobiles—are made up of multiple components. This dissertation comprises three essays that study the decisions of system innovators in mobile computing to “open” development of their systems to outside suppliers and the implications of doing so.

The first essay considers this issue from the perspective of which components are retained under the control of the original innovator to act as a “platform” in the system. Based on detailed review of leading systems in this industry in data spanning 1984-2004, I find that platform boundaries expand and contract over time as a means for platform suppliers to promote the adoption of the platform (by end-customers, downstream manufacturers, and suppliers of complements) and simultaneously to ensure the continuing innovation of the system.

The second essay provides a systematic empirical investigation of how the extent of openness chosen by the platform supplier affected the rate of innovation of mobile computing devices. I measure openness in relation to both the boundaries of the platform and the extent to which actions were taken by platform suppliers to promote entry by hardware device manufacturers. I find regular relationships across multiple measures of innovation that suggest that the whether openness increases or decreases innovation depends on the nature of innovation and the intensity of competition between device makers.

The third essay moves from device hardware to the applications software developer networks that form around opened platforms. I investigate the effect of platform suppliers’ aggressive promotion of entry of software developers around their platforms. I find that while large developer networks are associated with a wide selection of software titles, that very large developer networks in fact lead to less software output. I interpret this result as revealing the importance of maintaining investment incentives—even in a context with network effects.
Thesis Supervisor: Rebecca Henderson
Title: Eastman Kodak LFM Professor of Management

Thesis Supervisor: Michael Cusumano
Title: Sloan Management Review Professor of Management

Thesis Supervisor: Marc Rysman
Title: Assistant Professor of Economics, Boston University
Contents

Acknowledgments 9

1 Introduction 13
  1.1 Open Questions on “Openness” .......................... 14
  1.2 Dissertation Overview ................................. 16

2 The Boundaries of the Platform 21
  2.1 Introduction ........................................... 21
  2.2 Are Platform Boundaries Different from Firm Boundaries? ................ 24
    2.2.1 Platform Boundaries and Solving Coordination Problems .............. 24
    2.2.2 Setting Platform Boundaries to Allocate Investment Incentives ........ 26
    2.2.3 Platform Boundaries and the “Power” to Regulate Other Suppliers .... 27
    2.2.4 Modular Systems, Specialization and “Best of Breed” Component Suppliers 29
    2.2.5 “Opening” a System, Appropriability, and Interactions with Systems
        Competition ........................................... 30
    2.2.6 Summary ........................................... 31
  2.3 Empirical Context and Data ................................ 32
  2.4 Psion & Symbian ......................................... 35
    2.4.1 Integrated Development and Quantum Leaps in Design .................. 37
    2.4.2 Integration Extension of the System to New “Layers” .................. 40
    2.4.3 Tapered Disintegration & Opening the System .......................... 41
    2.4.4 Shared Control over the Platform ................................ 45
  2.5 Palm ................................................... 47
2.5.1 Palm Pre-History: Failures in “Modular” Development ............. 48
2.5.2 Integrated Early Development—Despite Lack of Experience .......... 50
2.5.3 Tapered Disintegration & Opening the System .................. 53
2.5.4 The Separation of Platform from Hardware Businesses .............. 56
2.6 Microsoft ............................................................................ 56
  2.6.1 Early Autonomous Innovation on a Small, Standard Operating System ........ 57
  2.6.2 Failed Attempts to Adapt Existing Technologies to Ambitious Visions .. 59
  2.6.3 Integrated Development to Produce a Highly (Overly) Modularized System 61
  2.6.4 Increasing Integration, Tightening Specifications and Increasingly Successful Products ........................................... 64
2.7 MontaVista’s Mobilinux ......................................................... 65
  2.7.1 Serving Disparate Applications by Porting to Multiple Board-Level Platforms ................................................................. 66
  2.7.2 A Mobile Computing Platform Created with a Coalition of Specialized Partners .............................................................. 70
  2.7.3 Creating Forbearance and Coordination Among Interdependent Suppliers 72
2.8 Discussion ............................................................................ 73
2.9 Conclusions ......................................................................... 76
2.10 Appendix ........................................................................... 83

3 The Effect of Openness on Innovation ......................................... 85
  3.1 Introduction ......................................................................... 85
  3.2 Conceptual Framework & Hypotheses Development .................. 88
    3.2.1 Diversity ....................................................................... 89
    3.2.2 Investment Incentives ................................................... 89
    3.2.3 Coordination and Governance ....................................... 91
  3.3 Competition and Industry Dynamics in Handheld Computers and Smartphones .. 93
  3.4 Data Set ............................................................................. 97
    3.4.1 Measuring Innovation .................................................. 97
    3.4.2 Measuring Openness as Granting Access and Devolving Control .... 99
    3.4.3 Other Data and Variables ............................................. 100
# List of Figures

2-1 Trends in Market Size and Shares .............................................. 33
2-2 Key Component Distinctions in Mobile Computing Systems .......................... 34
2-3 Vertical Integration Trends of Psion and Symbian .................................. 36
2-4 Vertical Integration Trends of Palm ............................................... 48
2-5 Vertical Integration Trends of Microsoft (in Mobile Computing) ...................... 58
2-6 Vertical Integration Trends of Montavista ........................................ 67
2-7 List of Interviewees ................................................................. 84

3-1 Industry Sales, Product Releases and Market Shares .................................. 93
3-2 Innovation of Products in the Leading Systems ...................................... 95
3-3 Variable Definitions and Descriptions ................................................. 101
3-4 Two Way Correlations Between Main Variables .................................... 102
3-5 Restatement of Hypotheses in Terms of the Empirical Strategy ................... 103
3-6 Low-Coordination Innovations Regressed on Openness, System-Level Data .... 111
3-7 High Coordination Innovations Regressed on Openness, System-Level Data .... 113
3-8 Within-System Entry Regressed on System Openness ............................... 115
3-9 Individual Supplier Product Development Decision Regressed on Within-System Market Structure .......................................................... 119
3-10 Within-System Variation of Products Regressed on Openness .................... 120
3-11 List of Product Characteristics Measured ........................................... 132
3-12 Characterization of the Components in Systems .................................... 133
3-13 Variables used to Construct Access Measure ....................................... 134
4-1 Growing Numbers of Applications Software Around the Leading Platforms . . . 143
   How Smartphones Work Wiley Publishers ............................................ 145
4-3 Number of Applications in Point of Sale Database over Time ................. 149
4-4 Trends in the Size of Developer Network and Price of Software Applications . . 150
4-5 Wide Variety of Complements Supplied .................................................. 152
4-6 The Long Tail of Demand Across the Selection of Software Titles ............. 153
4-7 Relatively Symmetric, Small Suppliers of Software Applications with Little Ten-
    dency to Concentration ............................................................................ 154
4-8 The Decision to Release Multiple Titles has a Greater Impact on the Number of
    Titles than Does New Entry ................................................................. 155
4-9 Average Prices Charged by Developer Firms of Different Sizes .................. 156
4-10 Distribution of Unit Demand Across Developers by Rank ........................ 157
4-11 Churn of Top Developers (Where the Top 25 Come From) ........................ 158
4-12 Rate of New Title Releases Regressed on Size of Developer Network .......... 160
Acknowledgments

Rebecca Henderson, my committee chair, was generous in many ways in supporting this research and my training more generally. In my early years in the PhD program, the hours I spent talking with Rebecca in reading courses combined in unforeseen and potent ways with the lessons I was learning in my economics classes. In her orbit, I was also regularly exposed to the special caution and care with which Rebecca addresses tricky issues around innovation and slippery issues around firm heterogeneity. Should I absorb a small fraction of these virtues they will be among the most valuable things I take with me from MIT. I’d also like to thank Rebecca for her “tough love” and criticisms, and for the considerable intellectual autonomy she afforded me. I’m sure I’m stronger for it. Finally, I’d like to thank Rebecca for being forever gracious, warm and charming.

Michael Cusumano was the second member to join my committee. Reading Michael’s book with Annabelle Gawer (while taking courses in organizational economics and industrial organization) should be credited for stirring many of the questions I posed in this work. Even more than this, I would like to thank Michael for having an enormous energizing effect on the work. Michael has an the unusual gift for addressing questions of daunting complexity and somehow—through pragmatic planning and disciplined execution—repeatedly coming up with tantalizing findings. His instincts for making progress on this tough empirical terrain were always on track and invaluable. Michael was also generous with his time and comments, never holding back a thought, always willing to offer an opinion and even venture a guess. Michael’s understated deadpan humor and nervy spirit made him a lot of fun to work with and make him a pleasure to know.

If nothing else, this dissertation was an attempt to bring systematic data to bear on a
tough problem. Marc Rysman, the third person to join my committee was a perfect role model. His work combines gritty knowledge of institutional detail and ease with modern theory and methods, and the discretion of how to invoke these. In Marc I found an ideal sounding board for how to best go about trying to draw systematic inferences from data in a complex and dynamic industry. He was also willing to roll up his sleeves on numerous occasions to help me think about the econometric identification challenges. He is filled with intellectual gusto while being exceedingly affable and easygoing.

I would also like to mention several towering forces in my intellectual development at MIT, those professors whose instruction influenced me most profoundly. In the economics department these were Jean Tirole, Glenn Ellison, Frank Fisher, Bengt Holmström and Jerry Hausman. In strategy, apart from Rebecca’s guidance was the mind-expanding discussion of the field with Giovanni Gavetti at HBS. Of all of my teachers, Bob Gibbons deserves special mention for his treatment of organizational economics—riveting as a Hollywood thriller, and enduring as one’s favorite book. His teaching is a service to the field and to his students. Bob was also kind to offer criticisms of my early dissertation proposal.

My interactions with Simon Johnson and Olivier Blanchard as their research assistant, very early in the program, taught me a lot about the warts and bumps of empirical research—and how these might be overcome. Also, any time I sat in a room with Ernie Berndt or Iain Cockburn, I found myself learning more about empirical research. I would also like to acknowledge several scholars whose comments made an important impact on the essays contained in this dissertation and the papers to follow: Richard Schmalensee, Carliss Baldwin, Jay Barney, Fernando Suarez, Erik Brynjolfsson, Joshua Gans, Luke Froeb, William Kerr, Melissa Schilling and Andrei Hagiu.

Among fellow PhD students and post doctoral fellows I would like to thank single out Nicola Lacetera, Ramana Nanda, Rafel Lucea, Rodrigo Canales, Karim Lakhani, Luis Vives, Lourdes Sosa, Steve Kahl, Hazhir Rahmandad and Sinan Aral for unfailing support, feedback and friendship. Nicola deserves special mention for being the closest of confidantes and the best of friends. I cannot imagine this experience at MIT without him. Thanks also to senior students in the program—and especially Mathew Bidwell, Sarah Kaplan, Henrik Bresman, Jordan Siegel and Andrew von Nordenflycht—for their advice, guidance and example. Perhaps unbeknownst to them, David Hsu, Jeffrey Furman, and Pierre Azoulay, deserve as much thanks for the guidance
they provided as forerunners in the program and great examples to follow.

Life in a PhD program is filled with unforeseen contingencies and Knightian uncertainty. Thank god for Sharon Cayley, whose alternate mercy and herding duties keep PhD students moving forward. She is the walking history and beating heart of the program. The administrative staff on the 5th floor were also helpful in many ways. Deserving special mention for their patience with me are Loretta Caira, Pam Liu, and Erin Lewis.

Thank you also to my family for unending support during these past five years. I am lucky, honored and blessed to be the son and son-in-law, brother and brother-in-law, uncle, husband and father to the likes of you.

My greatest teacher and the person deserving greatest acknowledgement is my wife, Elizabeth. She knew me well enough to understand and to support this path. She endured many days in which my thoughts and time were elsewhere, devoted to my work (“you and me and Stata makes three,” I’ve often said). And while giving up much, she’s given me so much: Life with her is a rousing musical.
To Libby
Chapter 1

Introduction

“The problem was the industry wasn’t measured by who has the best selling personal computer or who has the most innovative technology. The industry was measured by who had the most open system that was adopted by the most other companies and the Microsoft strategy ultimately turned out to be the better business strategy”

- John Sculley, as appearing in “Triumph of the Nerds: The Rise of Accidental Empires” Public Broadcasting Service (airing June 1996)

“Telecommunications used to be a closed game... Now, openness reigns.... Networks must interconnect with those of competitors and users can plug in their own devices as they will. One result of this openness has been a lot of innovation.”

- The Economist (January 8 2005)

The problem of choosing a best way to organize production (Richardson 1972; Williamson 1975; Grossman and Hart 1986; Teece 1986; Becker and Murphy 1991) becomes particularly challenging in “systems” industries—such as automobiles, information technology, aerospace, consumer electronics, and media products—where multiple components come together to form the final product or service. Apart from their often extraordinary scale and complexity (Marschak 1962), these industries are also often affected by exotic market conditions such as network effects, winner-take-all dynamics and other issues that interact with organizational decisions. In cases where a system is first innovated by a lone innovator who maintains property rights over
the system or where there is a platform supplier with market power (often the original innovator), the decision of how to organize these industries often falls to that party. It is primarily from the perspective of this original innovator or platform supplier that the question of how to organize a system is considered in this dissertation.

Today, the menu of options of how to organize production of a system is understood using the metaphor of pursuing an "open" or "closed" strategy. This description has traditionally been used to describe relatively clear-cut situations, such as the development of a public telecommunications standard to be maintained by a public institution for all to use or a proprietary system to remain secret and used exclusively by its developer (closed) (Katz and Shapiro 1986, 1994; David and Greenstein 1990). On the basis of these stark distinctions, theory and experience have revealed that giving up property rights and guaranteeing the free entry and use of a technology can promote adoption by reducing the threat of "lock-in" or hold-up of buyers (Shapiro and Varian 1998), by allowing multiple suppliers and buyers to coordinate on the same technology (Cusumano, et al. 1992; Grindley 1995; Augereau, et al. 2005), and by increasing mix-and-match benefits to users available in multi-vendor systems (Matutes and Regibeau 1988; Baldwin and Clark 2000). Of course, giving up property rights and allowing outside suppliers to enter also leads an innovator to lose its monopoly over the technology (Katz and Shapiro 1996; Economides 1996; Kende 1998). Given such high costs in lost appropriability, both research and practice has traditionally focused on the strategic decision to open a system in extreme cases of winner-take-all competition, where an innovator might be moved to pursue a "large share of a small market" in the case of a closed system or a "small share of a large market" in the case of an open system (Shapiro and Varian 1998).

1.1 Open Questions on "Openness"

In the current usage, however, the metaphor of openness appears to have become woefully overburdened in describing and prescribing the organization of modern technology systems. Rather than clearly distinct cases of devolving property rights and free entry versus tightly

---

1In earlier literature the favored terms were "sponsored" versus "unsponsored" systems.
2The counter-argument, that an "unsponsored" technology may receive less strategic investments and promotion of adoption through subsidies (Katz and Shapiro 1986) receives considerably less attention.
held property rights and foreclosed entry, “open” now casually describes a long list of issues in technology design, contracting and ownership choices. Is a computer language that is widely distributed and able to interact with multiple computer platforms—but over which ownership is maintained—“open”? Is a royalty-free music standard that forces conformance to certain technical rules and digital rights management “open”? Is an operating system that is informally governed, but governed nonetheless by an elite set of programmers “open”? Is a public communications network that has a government as benefactor and keeper truly “open”? Is a good sold at zero price “open,” independent of other considerations? Is a system with some, but not all, components opened while others remain proprietary really “open”? These scenarios suggest there might be different modes and degrees of openness—or simply there are multiple relevant technical and contractual instruments through which systems industries are organized and governed. Whether the application of these instruments is or should be so correlated as to allow a simple description along the continuum of open to closed remains unknown and under-explored.

Apart from the metaphor and its meaning, the theory also appears strained. Numerous scholars have begun to present evidence suggesting that openness also affects the continuing technical change and innovation of a system. Given the radical reorganization of production implied by opening a system, it perhaps should not be surprising that the opening of a system might affect the rate—and possibly direction—of technical change in a system. Perhaps more surprisingly, the bulk of the empirical findings seem to suggest that openness tends to have a positive effect on innovation (Langlois 1992; Baldwin and Clark 2000; von Burg 2001; Chesbrough 2003; von Hippel 2005). Might there be instances in which openness in fact hurts innovation? How would we know? Would it depend on the type or mode of openness in question? These issues also remain under-explored.

3 West (2004) surveys several of the issues.
4 Systems industries are often associated with an on-going stream of new products and components embodying novelty, technical advance and declining costs (Langlois and Robertson 1992; Bresnahan and Greenstein 1999; Cusumano and Gawer 2002)
5 Early theoretical models on the topic suggest a more contingent view of the effect of openness (Farrell and Katz 2000; Becchetti and Paganetto 2001; Nahm 2004).
1.2 Dissertation Overview

This collection of three empirical essays attempts to make progress in understanding how systems industries should best be organized by empirically studying the mobile computing industry (i.e. handheld computers and smartphones). The relatively clear-cut and comparable structures across multiple systems and years in this industry provided a unique opportunity for building large scale datasets. The analysis largely takes the perspective of the original innovators cum platform suppliers, interpreting the rationales and effects of their attempts to arrange the structure of production so as to create value, appropriate a fraction of that value, and allow the system to compete successfully against other systems in the market.

The first essay provides a rich historical description of the motivations and context for leading platform owners to give up control of different components in their systems at different times. In this essay, I focus on a most basic definition of openness—control over individual components in the system by the system innovator cum platform supplier—and study the motivation for retaining or giving-up control over these components (and, by implication, the boundaries of the platform). I find that shifting platform boundaries reflect attempts to negotiate tensions between goals of 1) retaining appropriability over the system; 2) promoting network effects and adoption dynamics; and 3) promoting innovation. The promotion of innovation sometimes involved closed strategies to achieve greater coordination; in other instances open strategies promoted external innovation.

The second essay provides a systematic investigation of the relationship between the extent of openness of systems and innovation outcomes with detailed data on the advancing attributes of hardware devices released in competing systems. In this paper I argue that increasing the openness of a technology—in terms of devolving control and granting access—should create tensions between promoting investment incentives, creating diversity in the supplier pool and promoting the efficacy of the coordination regime. The openness-innovation relationship is studied using a novel panel data on multiple dimensions of innovation in handheld computers and smartphones built "on top" of competing platforms of varying degrees of openness.

Prior empirical work on the supply-side of systems industries has relied on extensively detailed case studies, likely a result of limited numbers of competing systems, complexity of industry structure being studied and practical data collection limitations.
sistent with predictions, I find the sign and magnitude of the openness-innovation relationship depends on the type of innovation in question and the nature of strategic interactions between entering suppliers. Analysis of underlying mechanisms further corroborates the interpretation of trade-offs involved in opening a technology.

The third essay attempts to more closely examine the economic mechanisms underlying the granting of access to a component by studying the effect of promoted entry into third-party software applications development. Fantastic numbers of software developer firms—thousands—entered in a pattern consistent with network effects acting between increasing entry of software developers and supply of software titles, and an increasing installed base of end-users. However, close examination of the data suggest that despite the presence of network effects, that growing entry did not have nearly as much effect on the overall selection of software titles offered to end-users as did the discretionary development decisions by individual developers. The analysis finds that larger developer networks in fact lead to fewer software titles generated. I interpret this result as consistent with diminished investment incentives of incumbent developers with the growing competition created by incremental entry to developer networks.
Bibliography


Chapter 2

The Boundaries of the Platform

2.1 Introduction

"Systems" are products made up of multiple components that must work together. Examples include computers, software and telecommunications systems. A subset of components in a system used in common or reused across multiple products is a "platform." Typical examples include computer operating system software and the frame, suspension and drive-train used across multiple automobile models.¹

For the most part, research on economics and strategy in systems industries begins with assumptions of which components are part of a platform and which are not, and then concentrates on studying strategic interactions between the platform supplier, other component suppliers and end-users (Church and Gandal 1992; Armstrong 2002; Tirole and Rochet 2003; Rochet and Tirole 2005). However, recent empirical accounts suggest considerable plasticity in platform boundaries and the possibility that platform boundaries—or, rather, the boundaries of the platform supplier—might be responsive to economic incentives (Gawer and Cusumano 2002; Evans, Hagiu et al. 2004; Gaver and Henderson 2004). The active corporate strategies of today's prominent platform suppliers² gives further suggestion of a possible strategic role for

¹Beyond high-technology systems, a wide variety of contexts such as shopping malls and store-tenants, newspaper advertiser-reader relationships, and restaurant franchisor-franchisee relationships might be likened to systems in conforming to these general definitions.

²For example, Ebay, operator of a Internet auction platform, recently purchased Internet-telephony service provider, Skype; Google, provider of an advertising-search platform, recently acquired telecommunication fiber backbones; Microsoft, maker of an operating system platform, has long also developed application software.
the platform supplier's integration decisions.

Several papers have begun to elucidate a number of possible tensions that a platform supplier faces when choosing whether to "enter" (vertically integrate) into markets for complementary components (Church and Gandal 1992; Farrell, Monroe et al. 1998; Davis and Murphy 2000; Farrell and Katz 2000; Becchetti and Paganetto 2001). These papers suggest, for example, that widely integrating across a system might improve multi-market pricing of complementary components, allow "control" to be exerted over the system, and allow the platform supplier to extract a larger fraction of profits in the system. A number of complementary views on the boundaries of the platform might also be discerned from other analyses of industry structure in systems industries (Katz and Shapiro 1994; Baldwin and Clark 2000). The treatment of these issues in terms that are often idiosyncratic to systems industries and the frequent couching of these decisions as problems of entry (rather than, say, property rights as do many modern theories of the firm) prompt the question: Are the platform boundaries somehow different from those of other firms?

This paper attempts to contribute early steps towards a greater understanding of the determination of platform boundaries by offering a detailed descriptive analysis of vertical integration of multiple mobile computing platforms from the mid-1980s through to the mid 2000s. The context provides a cross-section of platforms whose vertical scope varied over time and in relation to each another. Five market-leading platforms were studied closely (Palm, Psion, Symbian, Microsoft and MontaVista embedded Linux) and six other platforms are also reviewed (platforms of Apple, General Magic, Geoworks, GO Corp., Casio and Sharp). The analysis proceeds as a descriptive account of patterns of vertical integration in each case, based on detailed longitudinal databases that were developed to summarize vertical scope in each system. The interpretation of trends in vertical integration was based primarily on a comprehensive review of press releases of the platform suppliers, periodical articles, and books written on the industry. These materials were supplemented with interviews with managers in the industry. Guiding this analysis, five alternative views of platform boundaries, not only from literatures on systems but

browser software, and even produces hardware with its Xbox; PalmSource, maker of the Palm OS mobile computing operating system recently announced it would port its platform to the Linux operating system kernel, effectively disintegrating from its own kernel development; disintegrated from operating system kernel development; large auto makers now themselves make a smaller portion of the automobiles they sell than ever before.
also relating to modern theories of the firm more generally, were reviewed and corresponding hypotheses developed.

**Summary of Main Findings** The simple descriptive facts of this industry are themselves noteworthy. In great contrast to ideas of a static and defined platform (such as "operating-system-as-platform") the analysis finds considerable variation over time and across systems in the level of integration of platform owners. Platforms were sometimes "wide," sometimes "thin;" sometimes expanding, other times contracting. And rather than discrete decisions of entry into a particular category of business (such as the decision of operating system developers to enter into the production of hardware devices), integration decisions were more involved more incremental winnowing or stretching of boundaries in relation to outside suppliers of complementary components. Also in contrast to notions of there being a clear correspondence between a "platform" and a "platform supplier," there were also cases where the technical platform (those components used in common) were supplied by firms other than just the ostensible platform supplier. For example, it has become increasingly common for graphical user interfaces and software development tools (both components used in common by multiple suppliers in a system) to be supplied by firms other than the ostensible platform supplier.

The overall logic of vertical integration that is found in these data reflects a combination of predictions from both the systems literature and the New Institutional Economics, and can be understood as reflecting tensions between several objectives:

- **Maintaining appropriability** by integrating into a lucrative component;

- **Promoting network effects** by encouraging outside firms to enter into a system, with the presumed effect this would then stimulate end-user demand;

- **Fostering innovation** by sometimes either integrating to promote coordination of difficult system-wide innovations, or sometimes opening a component to exploit the varied experience of outside suppliers (while simultaneously possibly promoting network effects and stimulating competition between these outside suppliers).

The paper proceeds in Section 2 with a review and synthesis of prospective theories of platform boundaries and highlights the alternative predictions of these views. Section 3 then
reviews the empirical context of the mobile computing industry and discusses the data used and how it was collected. Section 4 reviews the descriptive histories of the platforms to discern the motivations for vertical integration decisions in each case. Section 5 discusses the findings and concludes.

2.2 Are Platform Boundaries Different from Firm Boundaries?

From a technical standpoint, a product platform is a subset of components in a system that is used in common or reused across multiple products (Meyer and Lehnerd 1997). They are also often "foundational" in the sense of serving as a basic input for subsequent development, inevitably defining essential features of the products built upon them. Thus, components forming part of a platform tend to possess some sort of economies of common use, such as economies of scale or economies of standardization. For example, the benefits of standardization in operating system software often outweigh the benefits of diversity; for applications software it is often the opposite case. On the basis of these reasons, a platform becomes the "one" in a "one-to-many" set of technical relations with other components. Further, the platform might be said to "enable" product development "in the sense that platform reuse may imply lower fixed costs of the development of any one product built "on top" of the platform. However, technical structure alone does not provide a sufficient basis for understanding the economic division of production by firm boundaries (Williamson 1971).

To develop hypotheses of how the characteristics of systems and platforms might interact with the logic of economic organization, several modern perspectives on the theory of the firm are reviewed along with views on platform boundaries directly taken from literature on systems competition.

2.2.1 Platform Boundaries and Solving Coordination Problems

Transaction Cost Economics (Williamson 1975) originated with the idea that for whatever inefficiencies integrated firms may or may not suffer in relation to market coordination, they will have privileged access to instruments of coordination, and thus activities involving pro-

---

3 Later developments in this literature came to criticize this seemingly one-side view of the benefits of firms (Grossman and Hart 1986), but for our purposes in this paper it will be useful to preserve this original interpre-
nounced coordination problems should be brought within the boundaries of a single firm. Interpreted broadly, integration might solve coordination problems emanating from a wide variety of sources—asymmetric information, economic spill-overs, investment externalities, simple coordination of the left-hand-knowing-what-the-right-hand-is-doing kind, and others. The Transaction Costs Economics literature has most often emphasized costs of “hold-up” and “haggling” that can arise where there are appropriable rents in a relationship on the basis of parties having made relationship-specific relationships in an environment where it is not possible to write complete contracts (Williamson 1975; Klein, Crawford et al. 1978).

Research on systems similarly recognizes virtues of integration usually described as the advantages of maintaining “control” over the system (Katz and Shapiro 1986; Becchetti and Paganetto 2001). The ability of a single integrated firm to implement better coordination has been raised in the contexts of attempting to achieve coordinated and “coherent” advance of a system across multiple components (Bresnahan and Greenstein 1999; Gaver and Cusumano 2002). Large, integrated suppliers of systems are also considered to have inherent advantages in coordinating and developing the necessary technical standards involved in ensuring the inter-operation of components (Farrell and Saloner 1985; Besen and Farrell 1994). The view of autonomous component suppliers failing to coordinate is perhaps revealed most starkly in the treatment of effective collusion on pricing decisions, such as multi-component pricing decisions (Davis and Murphy 2000; Davis, MacCrisken et al. 2001) and bundling decisions (Bakos and Brynjolfsson 1999; Carlton and Waldman 2002), where coordinated pricing is characterized as only possible in integrated firms.

In the extreme, where maintaining coordination is the over-riding concern across an entire system, the entire system may be produced by an integrated firm. More generally, components in the system involving strong technical interactions, economic externalities or other potential sources of coordination failure should be brought within the boundaries of a single firm. How-

---

4 Based on the early conceptions of this literature, placing production within firm boundaries will benefit from, for example, the ability to invoke fiat in resolving conflicts; the interests of parties within a firm may be more closely aligned than those in different firms; cooperation may be better sustained by the threat of alienation; and firms may possess a wider variety of control instruments (Williamson 1971). Numerous other theories of the firm offer complementary perspectives on why firms may have advantages in solving coordination problems (ex: Kogut and Zander 1996).

5 Integration in these cases might also minimize anti-trust scrutiny (Tirole 1988).
ever, Transaction Cost Economics does not, at least on its own, militate the integration of all coordination problems be carried out by the platform supplier. For example, a coordination problem in developing two hardware components need not necessarily be accomplished by integration of an operating system developer. Therefore from this perspective, the production of components in the system involving high technical interactions, economic externalities, mutual specialization or other sources of potential coordination problems will tend to be integrated within a single firm.

A possible difference between platform components and other components, however, is the likely severity of coordination problems, leading to more likely integration of all components of a technical platform by a single platform supplier. For these components that serve as the basic, system-defining, components, there is more likely to be a great deal of specialization of the components to one-another. Indeed any specificity of investments is likely to be in relation to these system-defining components. Further, high-fixed costs and economies of common use would likely lead to few suppliers of each individual platform component, increasing the hazard of bilateral hold-up problems. For example, separating the development of an operating system from the application programming interface (API) library of instructions that is used by programmers would create enormous mutual dependencies between the supplier of these separated components. Therefore, from this perspective, the production of components used in common in a system, the “platform,” may tend to be supplied by a single firm, the “platform supplier.”

2.2.2 Setting Platform Boundaries to Allocate Investment Incentives

Property Rights theory (Hart 1995) is closely allied with the preceding Transaction Cost Economics perspective in emphasizing the need to make relationship-specific investments and hazards thereof. In this view, however, the firm is not a privileged institutional setting for achieving coordination. Rather, both cases of integration and disintegration must equally face the problem that bargaining (or worse, haggling and hold-up) over appropriable rents can dimin-

---

6 Numerous variants of original idea of using firm boundaries and the allocation of residual rights of control to mediate investment incentives (and recently, an emphasis on the ability to efficiently adapt to unforeseen contingencies) have been elaborated since the original property rights view was developed. This paper simply wishes to highlight the use of shifting boundaries to mediate investment incentives in broad brush strokes.
ish the incentives of trade partners to make (uncontractable, relationship-specific) investments. And so, firm boundaries around the physical assets of production are set to give control to the party whose (marginal, uncontractable, relationship-specific) investments are most important (Grossman and Hart 1986; Hart and Moore 1990).\(^7\)

The idea of allocating property rights to promote investment is integral to research on systems competition. Most often, the investment decision considered is that of outside firms to enter or adopt a system,\(^8\) where a platform supplier can promote entry (investment) by devolving control over a system and thereby reduce the risk of hold-up (or “lock-in” in the words of the systems literature). This becomes particularly relevant in systems contexts where bargaining power is inherently skewed towards the single supplier of a platform in relation to possibly many complementary component suppliers. The system literature most often emphasizes hold-up as a strategy, with phrases as “commoditizing the complementors” or “lock in” of the “installed base” (Shapiro and Varian 1998; Shapiro and Varian 1999). However, oppositely, retreating boundaries of the platform (Farrell and Katz 2000; Gawer and Henderson 2004) and devolution of property rights (Katz and Shapiro 1994) have been argued to promote investments of trade partners.\(^9\) Therefore, from this perspective, The boundaries of the platform supplier will shrink in relation to those suppliers whose (marginal, uncontractable, relationship-specific) investments are relatively important. Further, the platform supplier’s boundaries will generally increase where its own (marginal, uncontractable, relationship-specific) investments are relatively important.

### 2.2.3 Platform Boundaries and the “Power” to Regulate Other Suppliers

Agency Theoretic perspectives emphasize inefficiencies arising when one party (a “principal”) gets its work done through other parties (the “agents”) (Holmström 1979). To minimize agency

---

\(^7\)Optimal asset allocation may become more nuanced when other factors are considered such as long-term relationships or “repeated interactions” (Halonen 2002; Baker, Gibbons and Murphy 2002).

\(^8\)Adopters of products may need to make relationship-specific investments in learning, say, a particular computer language associated with a platform; suppliers of components working with a platform may need to invest in platform-specific designs, plants and skills. In the presence of network effects, there is also effectively a “collective switching cost” (Farrell and Klemperer 2003), whereby adopters become locked-in to a relationship by force of the tipping of the market to a given platform.

\(^9\)Bargaining power and investment incentives might also be modulated through institutional arrangements (public standards organizations, reputation) or by credibly committing to trade only with a small number of suppliers (Brynjolfsson and Bakos 1993; Stole and Zwiebel 2003).
costs, a principal will devise a system of incentives and constraints to attempt to regulate agents’ behavior (Holmström and Milgrom 1991). A powerful principal at the nexus of multiple such principal-agent relationships will have further enhanced abilities to reduce agency costs (Alchian and Demsetz 1972) by invoking multiple incentive instruments in nuanced fashion (Holmström 1999) and by regulating agents’ behavior in ways that contemplate the externalities agents’ actions may exert on one another (Segal 1999). An important basis of this power is ownership over the critical assets of production (Hart and Moore 1990; Holmström 1999), as agents will accede to terms of access where the assets make them more productive. The right of exclusion conferred through such property rights will also allow the owner-principal to directly regulate the identify and number of agents (Rajan and Zingales 1998). Thus, firm boundaries may be established so as to consolidate control over critical assets of production and in this manner gain the ability to “regulate the subeconomy” around those assets by devising an “incentive system” (Holmström 1999).

Research on systems suggests an analogous role played by platform suppliers at the center of business groups supplying systems. Platform suppliers have been likened to “licensing authorities” (Rochet and Tirole 2005) with the power to withhold access to the system through control over the technical platform, and on this basis they too have the ability to set the terms of access. The extent to which the innovator can maintain control over critical elements of the system may then have the benefit of allowing the platform supplier to exert some indirect influence over the entire system in a manner akin to a “public regulator” (Farrell and Katz 2000) or “regulatory commission” (Rochet and Tirole 2005). Further, where retaining control over critical assets implies retaining appropriability the innovator is more likely to possess the incentives and resources to ”sponsor” the system, making common investments, and strategic

---

10 A principal at the center of trade relations may also have advantages in holding a reputation (Williamson 1983), possessing “focalness” (Kreps 1990), serving as a central conduit for information flow (Marschak and Radner 1972).

11 A liberal interpretation of the original text.

12 Principal-agent models do not in themselves directly suggest a theory of asset allocation and firm boundaries.

13 The analogy may not directly apply to problems of asymmetric information or moral hazard particularly, but more generally regulating actions beyond the principal actor’s own boundaries.

14 Recent research attention in has been paid to how pricing terms can be set to exploit complementarities across parties adopting or entering the system (Tirole and Rochet 2003; Evans and Schmalensee 2005). However, empirical evidence and preliminary theory suggests a wider variety of non-price terms are also set by a platform supplier (Gawer and Cusumano 2002; Rochet and Tirole 2005).
subsidies (Katz and Shapiro 1986). In the case of a system product, "rules" or "laws" embedded in the technology (Lessig 2000) that determine the freedoms and restrictions and otherwise contour the costs of firms in the business group using the technology will add to this ability of the platform supplier to coordinate economic activity in the business group. The sum of the coordinating abilities of the platform supplier beyond its own boundaries has been referred collectively referred to as "platform leadership" (Gawer and Cusumano 2002). Therefore, from this perspective, the boundaries of a platform supplier will tend to encapsulate critical components that confer the ability to regulate entry or otherwise directly set the "rules" of conduct in the system.

2.2.4 Modular Systems, Specialization and "Best of Breed" Component Suppliers

The Modularity perspective on systems organization begins with the idea there can be advantages to decomposing a product into a system of substitutable components, interoperating through stable interfaces within a predefined over-arching architecture (Baldwin and Clark 2000). Despite a priori advantages of "global" design approaches that do not fix a set of design decisions before the fact (as is done in fixing architecture and interface rules), an approach may make a complex design task more tractable (Simon 1973). However, where the task of decomposing a product into modules is itself a complex task, modularization may itself only emerge from a gradual search process (Clark 1985; Baldwin and Clark 2000; Ethiraj and Levinthal 2004).

Beyond a motivation for specialization and the division of labor, modularity may also bear on firm boundaries by acting on the magnitude of coordination problems between suppliers, discussed in earlier subsections. Modularization will, for example, reduce the extent of cross-component technical interactions and thus reduce the need for explicit coordination. Further, where codified interfaces separate not just substitutable components but substitutable component suppliers, investments may become less specific to any one bilateral trade relationship and thus mitigate bilateral hold-up and under-investment problems. In this sense, greater modularity may allow greater disintegration simply by allowing it to take place with less practical costs (Baldwin and Clark 2003). Therefore, the boundaries of a platform
supplier are more likely to winnow as the underlying design of a system matures and becomes increasingly modularized.

Where modularity reduces the transaction costs the importance of relative efficiency differences among firms may be come more pronounced. For example, a diverse set of suppliers may be better able to generate a variety of components that lead to mix-and-match advantages for users (Matutes and Regibeau 1988). Individual specialized component suppliers might also be better able to deliver superior “best of breed” components that can be brought together within a modular system. This may be especially important where there are strong economies of scale or advantages to specialization in the production of individual components (Stigler 1951), where the vagaries of technical advance make it difficult to know which component designs will be superior before the fact (Garud and Kumaraswamy 1995), or where there is significant heterogeneity in firm resources, comparative advantages and productive knowledge (Chesbrough 2003). Therefore by reducing the scope of the platform supplier to include fewer components, the system may benefit from diverse suppliers or suppliers of superior production efficiency. Therefore, this perspective also suggests the boundaries of the platform supplier will tend to shrink in relation to those components where the platform supplier does not hold a monopoly over production knowledge and efficient production.

2.2.5 “Opening” a System, Appropriability, and Interactions with Systems Competition

The earlier discussion of the Allocation of Investment Incentives pointed out that reducing the vertical scope of a platform can promote entry, adoption and investment in the system. This devolution of control is of special interest in research on systems industries where competition may take be of a winner-takes-all kind: small early advantages can lead to virtuous circles of entry, adoption and investment and market domination of one system over all others (Katz and Shapiro 1986; Katz and Shapiro 1994; Shapiro and Varian 1998; Bresnahan and Greenstein 1999). From this perspective, the particular components that are first opened will be those with the greatest likelihood of fomenting these sorts of positive feedback. For example, indirect

---

15 These benefits might also be gains through tapered disintegration, where the platform supplier does not disintegrate from the production of a given set of components, but allows other suppliers to provide substitutes to the components it supplies.
network effects between variety in applications software and hardware adoption may lead the platform supplier to disintegrate from software supply (Church and Gandal 1992). Therefore, the boundaries of the platform owner will tend to shrink in relation to those components that are most likely to foment network effects or other such virtuous circles of adoption, entry and investment.

Research on systems, however, stresses the possibility that the platform supplier may be reluctant to set the level of “openness” in a way that maximizes surplus creation and the competitiveness of the system—and only under the duress of weakened competitive position of the system will the platform supplier open the system and trade-off “adoption and appropriability” (Shapiro and Varian 1998). The implied reasoning is that high uncertainty, complexity and practical contracting limitations make it impossible to receive full discounted value of profits deriving from opened components from entering component suppliers. Where devolving property rights sacrifices appropriability and ex ante contracting on transfers of property rights is not possible, we might also expect a general reluctance of the platform supplier to disintegrate. And so, platform boundaries will tend to remain integrated in components of high value, that preserve appropriability, that will only be disintegrated under intense competitive pressure.

2.2.6 Summary

The preceding discussion reveals a range of possible motivations for a platform supplier to actively manage its economic boundaries. The following table roughly summarizes these points. The theories of the firm considered here emphasize coordination and investment problems of various kinds and how regulating the boundaries of the platform might solve these problems.
<table>
<thead>
<tr>
<th>Theoretical Perspective</th>
<th>Integration</th>
<th>Disintegration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theories of the Firm</td>
<td>Direct coordination</td>
<td>Incentives for component suppliers</td>
</tr>
<tr>
<td>Transaction Costs</td>
<td>Incentives for platform supplier</td>
<td></td>
</tr>
<tr>
<td>Property Rights</td>
<td>Indirect “regulation”</td>
<td>Component supplier heterogeneity</td>
</tr>
<tr>
<td>&quot;Power&quot; and</td>
<td></td>
<td>Network effects</td>
</tr>
<tr>
<td>Incentive Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theories of Systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modularity</td>
<td>Appropriability</td>
<td></td>
</tr>
<tr>
<td>Open Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 1: Alternative Motivations for Adjusting Platform Boundaries*

Theories of platform boundaries implied by research on systems were consistent with these theories of the firm, but emphasized the relationships between coordination problems and the underlying technical design, productivity differences between outside component suppliers and the platform supplier, and interactions between vertical integration, market competition between systems and appropriability. These points appear to be extensions consistent with the existing theory of the firm rather than departures from it.

2.3 Empirical Context and Data

After commercial beginnings in the 1980s, the market for mobile computers began to take-off in the 1990s, as shown in the following figure. In early periods, the "palmtop" design dominated sales, with Psion taking early industry leadership. Psion’s leadership was then broken by the entry of Palm Computing in 1996, with the launch of its popular PDA, the PalmPilot. Struggling to maintain its market position, Psion soon thereafter transformed to Symbian with fresh funding from mobile phone manufacturers. Eventually in the 2000s, Microsoft’s mobile computing platforms rose in prominence with its own PDAs (after over a decade of failures to generate compelling technology). As the bulk of the market opportunity and growth shifted from handheld computers to smartphones (built, in large part, on the same technical platforms)
the renovated and revitalized Symbian platform (re)emerged in as a market leader. A cadre of other firms entered attempting to develop new systems that could exploit this new opportunity. Notable among them were Research in Motion (RIM), and a variety of Linux distributions tailored for mobile phone devices.

**Vertical Integration Database** Data on platform boundaries were systematically compiled across leading mobile computing systems on a quarterly basis. Data were collected on both the succession of market leaders in the industry (Psion, Palm, latter-year Microsoft, Symbian), high profile failures (early-year Microsoft, GO Corporation, Apple Newton, and several others), and up-and-coming Linux distribution supplier, MontaVista. A database of vertical integration patterns over time in each of these platforms in relation to the components, listed in the following figure, was developed for each quarter of each year since the original development of these systems. The database was built on the basis of a comprehensive review of press releases by platform suppliers of interest and their main trade partners, and comprehensive review of all articles in relation to the platform supplier and the mobile computing industry that appeared in the Lexis-Nexis database from the original development of these systems. The data are summarized graphically in figures presented within the analysis, itself.
<table>
<thead>
<tr>
<th><strong>SYSTEM COMPONENTS</strong></th>
<th><strong>SUB-COMPONENTS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>APPLICATIONS</strong></td>
<td>Software designed to accomplish a specific task, such as desktop publishing or word processing</td>
</tr>
<tr>
<td></td>
<td>Applications</td>
</tr>
<tr>
<td></td>
<td>Integ'd Dev't Env't (IDE) &amp; Tools</td>
</tr>
<tr>
<td></td>
<td>Framework &amp; App'n Prog'g I/F (APIs)</td>
</tr>
<tr>
<td><strong>OPERATING SYSTEM</strong></td>
<td>(Set of) system memory-resident software that links applications programs and input-output devices to machine-level instructions</td>
</tr>
<tr>
<td></td>
<td>Graphical User Interface (GUI)</td>
</tr>
<tr>
<td></td>
<td>Utilities</td>
</tr>
<tr>
<td></td>
<td>Kernel</td>
</tr>
<tr>
<td><strong>DEVICE</strong></td>
<td>Collection of board-level design, system integration, industrial design tasks that yields a working device</td>
</tr>
<tr>
<td></td>
<td>High-level definition</td>
</tr>
<tr>
<td></td>
<td>Board-level Specs</td>
</tr>
<tr>
<td></td>
<td>Complete design and integration</td>
</tr>
<tr>
<td><strong>PROCESSOR</strong></td>
<td>(Set of) component(s) designed to perform basic arithmetic and logic operations As defined here, also radio signal processing, low-level control of basic components Receives instructions from the operating system</td>
</tr>
<tr>
<td></td>
<td>Drivers and configuration</td>
</tr>
<tr>
<td></td>
<td>Microprocessor</td>
</tr>
</tbody>
</table>

Figure 2-2: Key Component Distinctions in Mobile Computing Systems
Secondary Data, Databases and Interviews  Interpretation of these trends was supported by review of these same sources—press releases and an exhaustive review of articles appearing in the Lexis-Nexus database. In addition, several books and periodicals related to the industry, listed in footnotes, served as important references. Finally, where questions remained in interpreting the data, interviews with industry participants were conducted to clarify matters. A total of 42 interviews were conducted via telephone, Internet and in-person as a supplement to the structured data-gathering. These include interviews with platform suppliers, their suppliers and partners, and industry analysts. Interviews included those with executives at Palm, Symbian, Psion and MontaVista. Interviews with Microsoft managers were not secured. A list of interviews can be found in the Appendix.

2.4 Psion & Symbian

The over two decades of industry-leading development of Psion’s handheld computer system, and later transformation to the Symbian, reveals several primary guiding principles in the setting of platform boundaries.\(^\text{16}\) In the early days of development Psion used high levels of integration across the entire system to enable the cross-component design decisions that would make its system successful. It continued in this mold for over decade, successfully achieving repeated quantum leaps in design, consistent with perspectives on firms boundaries from the New Institutional Economics and notably the Transaction Cost Economics perspective.

However, opening of the Psion system in response to competition and eventual divestment of Psion’s platform assets to a consortium of device manufacturers reflects a Systems competition perspective. Early attempts to open both software and hardware development were in the interest of fomenting network effects. Failing to regain momentum in the market, Psion ceded much control to outside device suppliers to promote continued investment, transferring its platform assets to the Symbian consortium with divided ownership.

---

\(^{16}\)This section draws substantially from interviews with current employees of Symbian, as well as past employees of Psion. Additional important references included, “The History of Psion” by Steve Litchfield, http://3lib.ukonline.co.uk/historyofpsion.htm, downloaded in Jan 2004; “How Smartphones Work” Symbian Press and Wiley (2006); “Digerati Gliterati” John Wiley and Sons (2001). Data on the scope of the platform was compiled through exhaustive review of all Psion and Symbian press releases and articles written on the companies found in the Factiva database, a compilation of periodicals, during the period of study. Details of product features were drawn from the database described in Chapter 3.
Figure 2-3: Vertical Integration Trends of Psion and Symbian
2.4.1 Integrated Development and Quantum Leaps in Design

Working closely together—even side-by-side in a small 3,500 square foot London office—Psion’s small cadre of engineers and scientists designed one of the earliest mobile computers, the Psion Organiser, in 1984. This first product was an 8-bit, calculator-like, 2k RAM device with a standard single line LED display. The device was built around an 8-bit NEC chip, a standard part in many CP/M microcomputers of the day. Despite these modest specifications, the cigarette-pack sized device that was expandable with modular memory cartridges, and was designed with database capabilities. It was also programmable, with a built-in BASIC-like language that became known as Psion’s Organiser Programming Language (OPL) (and would continue to deepen and evolve in future decades of products). In typical use, the device would last a remarkable 6 months on a single 9 volt battery. This success in design was able to be achieved through extraordinary parsimony in software design, with shared code across operating system and applications and tightly tailored component drivers were designed hand-in-glove with the hardware, and pre-installation of this code in system memory. Extraordinary performance was therefore wrung from the modest component technologies available at the time through design decisions with implications across components and throughout the system. The device sold a surprising 30,000 units with almost no marketing—years before most other computer firms began experimenting with mobile computer technologies.

Thus technical approaches and organization of the design team therefore reflected a need to make the best of limited performance of standard component technologies available at the time. The design task was essentially one of co-specializing mostly off-the-shelf components, a task that appears to have benefitted from the small, tightly-knit team, which included Ph.D.

---

18 All product characteristics in this section are drawn from the mobile computer product database described earlier in this paper.
19 Add-in memory cartridges were available, including those with mathematical and financial functions.
20 Psion had developed skills in highly efficient coding by, for example, producing flight simulator and business suite software for microcomputers in earlier years.
21 The ability to re-use code across applications, graphical user interface and operating system was a major tenet of Psion design throughout its history. Longtime Psion technologist and executive Colly Myers years later stated in a Psion press release: “Software re-use is a science for us, so the announcement of another two products utilizing EPOC further illustrates our platforms’ enduring and stable architecture.”
physicists and a small team of engineers who benefited from having had the experience of developing ultra-lean applications (including as games, a flight simulator, graphics and business suite software) for the Z88 architecture for the Sinclair microcomputer.  

In April 1986 Psion released a second version of its product, the Organiser II. With this product, Psion brought about quantum improvements to the design with a follow-up evolution of the product, using much the same approach. The Organiser II was designed with many more times the original system resources (and especially screen size and memory allocation), and a considerably more elaborate operating system and the ability to attach to peripheral hardware. The larger screen and greater memory resources also opened the possibility of running spreadsheets, word processors and diaries. The approach of maintaining an integrated approach to design appears to have again enabled these quantum advances.

The Organiser II quickly sold hundreds of thousands of units and won over large corporate buyers; the Organizer II and its successors would sell half a million units over the following decade. For example, large British retailers such as Marks & Spencer attached barcode scanning hardware to the Organiser II and used it in retail and logistics applications. So robust and successful was the design it remained in production for over a decade, retiring in the late 1990s.

In the midst of growing success of 8-bit Organisers, in 1987 Psion then began work on a then-radical 16-bit design, the Sixteen Bit Organizer (SIBO). Once again, Psion management deliberately decided to avoid “incrementalism” and to pursue aggressive design advances. The public offering of Psion stock on the London Stock Exchange in 1988 helped fund this development. The implementation of the complete vision of a handheld personal “palmtop” device would be achieved in several iterations. The first SIBO devices, the “MC” series (MC200, MC400, MC600), were released by late 1989 and were small sub-notebooks with long battery

---

22 The operating system itself began to take on much greater sophistication; David Potter referred to it as the “first real EPOC system” (Langdon and Manners 2001)
27 In relation, graphical video game consoles were still 8-bit at the time
life. This first set of devices did not implement the full graphical interface environment that was envisioned for SIBO devices, but already began to make advances on laudatory features such as long battery life (it would run for a week on six AA cells), a touchpad controller, slots for proprietary solid state (SSD) memory, docking bays for peripheral hardware, and voice interface and voice compression technologies.32

Despite the substantial technical advances of the MC generation of 16-bit SIBO, they sold just 2,000 units33 and nearly bankrupted the company.34 It was only with the radical shrinking of SIBO devices to become “palmtops” in September 1991 that new commercial success would arrive, beginning with the Series 3. These were pocket-able palmtop devices with full QWERTY keyboards, including a graphical user interface (years before Microsoft Windows would be available), with application suites that were comparable to desktop functionality (and continued to be extraordinarily efficiently designed and sharing code with the operating system35) and lasting for a month on two AA cells—for a fraction of the price of the products in the MC series. Several improved SIBO models (3a, 3c and several slight variants of these products) would be released by Psion in five years following, with updated industrial designs, more memory, software updates and new applications, longer battery life and infrared connectivity. These breakthrough designs led Psion to rise to market leadership in mobile computing, taking roughly one-third of worldwide mobile computer shipments at their peak of market share in the mid-1990s.36 Psion became part of the FTSE 100 of largest publicly-traded British companies. The creation of these palmtop devices would also begin a long string of awards for design.37 In 1994, amidst this success, Psion would repeat its prior pattern of implementing quantum advances in

---

37 Psion has won major awards throughout its history, including the high honor of winning several Queen’s Awards for export achievement (1995) and Technological Achievement (1993) for the Series 3a and several British Design awards. Other awards included “Best Consumer Product” from the Design Business Association.
beginning a three-year effort to completely re-write the platform with the goal of developing a future generation compact 32-bit multi-tasking, object-oriented operating system with support for a pen interface that would be known as EPOC32.38

2.4.2 Integration Extension of the System to New “Layers”

The commitment to integrated innovation continued in the expansion of the system, itself, to new layers and components of the system—most notably in networking and component technologies. Perhaps most aggressive was an attempt to bring integrated wired and wireless data and Internet solutions to both consumer and corporate markets. Working with Motorola, in 1994, Psion produced a network messaging and wireless data terminal service called RWAN. The RWAN device combined (then advanced) packet-data radio communications technology with Psion device, and entailed the development of E-mail gateways and host databases at the wireless carrier central office. Demand for the application never attained a sustainable scale. Working with Compuserve, Psion developed an dedicated Internet services for its palmtop users. To enable network communications, Psion developed 3Fax, a peripheral modem that would allow palmtop owners to dial-up Compuserve over the public telephone network and then send and receive e-mail and enjoy several services on a dedicated mobile server domain that were tailored to the palmtops.39 Psion also enabled networking to the PC desktop as it created its PsiWin synchronization software, rebuffing offers to use Palm Computing’s PalmConnect or other PC-synchronization software that was being adopted by numerous other mobile computer manufacturers at the time.40 Instead, it preferred to acquire Palm Computing’s software, but the parties could not agree on an acquisition price.41

Psion also attempted to integrate into development of advanced component technologies. For example, in the early 1990s Psion worked closely with Intel and SanDisk on what would eventually become industry standard compact flash solid state memory. In another show of the direction of the future market, Psion also attempted to acquire chip designer Acorn—the

---

39 Psion’s modem division, Dacom, simultaneously became a leading vendor of PC card modems in this period.
40 De Vries, A., I. Herrod, et al. (2004). Technology and Competition - Psion’s failure to introduce a tablet PDA.
company that would eventually provide chips for the Apple Newton and provide the de facto standard architecture in most modern day handheld computers and smartphones. The deal, however, fell through as the parties were not able to agree to a price. In an analogous inclination to make advances itself, years later in 1998, Psion led attempts to port a Java runtime machine to its platform.

These extensions to the core design—along with laudatory design features in the basic device designs themselves—each reflected an enviable ability of the Psion design team and its management to see future trends and winning technologies. Unfortunately, most of these technologies were innovated too early—both in the sense of the maturity of the market to adopt these technologies (as in mobile data and Internet access), or in the sense of the level of advance in price and performance that could be achieved with then-available technologies (as in speech recognition).

2.4.3 Tapered Disintegration & Opening the System

An exception to this trend to integration first appeared in applications software. Applications software was the earliest layer of the system to be “opened.” From the first Organiser in the 1980s, Psion devices had been user-programmable with the BASIC-like OPL language. Developer kits had been available for the MC series, but had arrived late and did not grant third-party programmers fully freedom to exploit the capabilities of the system. However, cues from the PC industry at the time suggested that the development of complementary software by a large network of outside suppliers could promote adoption of the devices. The stabilization of the SIBO platform in the early 1990s would also make this feasible. Thus, in the early 1990s

---

13 Acorn’s ARM chips were a revolutionary new architecture based on a reduced instruction set, resulting in dramatic advances in power efficiency and cost.
14 Psion’s development team began work on integrating a Java runtime engine as it complete development of its first release of EPOC32 in 1996—well before standard Sun-sponsored standard Java configurations were ready for commercial use (i.e. PersonalJava and, later, J2ME). Tasker, M. (2000). Professional Symbian Programming. Birmingham, UK, Wrox Press.
15 Users could program simple scripts that would be saved on the devices, themselves, using a simple Psion-developed BASIC-like language, called OPL.
17 Interview with George Grey, former President Psion, Inc USA: “This was the time when everyone was talking about compatibility and the lessons of the PC model versus Apple were becoming apparent. Really, if you were going to be successful you were going to have to openly license the platform. And it struck a chord with Psion.”
Psion deliberately promoted the entry of outside application software developers, investing in development tools and increasing the power of the OPL language in drawing on the capabilities of the EPOC operating system,\(^48\) granting access to key programming frameworks,\(^49\) hosting coding contests, and by offering to co-market and facilitate distribution with developers. A year after the release of the Series 3, developer kits allowing programming in the more powerful C and C++ programming languages also became available, along with OPL, OVAL, Assembler and simple macro-writing.\(^50\)

Unfortunately, the high cost of the peripheral software cartridges used to deliver third-party products\(^51\) and the immaturity of tools and support provided at the time resulted in titles that were as expensive as they were uncompelling. Moreover, the highly uncertain and meager market for applications software for mobile computers created only limited interest among outside developers to learn specialized development skills and to develop for the platform. Therefore, despite a widening range of software titles in the early 1990s—especially those that brought some form of “content” to the existing database functions (such as wine lists, travel schedules, baseball scores, etc.)—no significant market emerged for these titles. On the basis of a long history of software development experience and intimate knowledge of the system, itself, Psion remained far and away the most important developer of applications for the SIBO platform. Taking advantage of the capabilities of the new platform, such as graphical user interface, Psion developed appealing applications including personal information management, word processor, database, spreadsheet and other programs. Richer development tools, cheaper memory media, the ability to transfer software via Internet and ability to program in OPL, C++ and Assembly language would lead to much greater development activity in the second half of the decade.\(^52\) But even then, the ability for Psion to more effectively re-use the low-level

---

\(^{48}\)The first OPL Software Development Kit (SDK) giving many utilities and macros for nearly full access to the SIBO operating system services, was released shortly after the release of the first palmtop Series 3, in 1991. When Series 3a was released in 1993, OPL was upgraded again but would then remain relatively unchanged until 1996 when OVAL (Object-based Visual Application Language) and associated development tools were created developed to be compatible with Microsoft Visual Basic. Development in more mainstream C++ would come relatively late and only with a new generation of platform in 1997.

\(^{49}\)Psion also allowed programmers access to use the database framework that underpinned Psion's own database and spreadsheet applications


\(^{51}\)While elegant and a harbinger of future technologies in the industry, a simple cartridge without software was priced at $100 in the early 1990s. Cartridges with pre-loaded software sometimes exceeded $200.

\(^{52}\)Steve Litchfield asserts that eventually 1500 3rd-party titles would be written for the more popular Series
code base made the company the most productive software developer.\textsuperscript{53}

Psion executives were far more ambivalent in their decision of whether to open the hardware business. Hardware was the company's primary source of income. It was only following the precipitous decline in Psion's worldwide market share in 1996 resulting from the launch of Palm Computing's wildly popular Palm Pilot that Psion chose to finally open the hardware business to outside suppliers\textsuperscript{54} (although Psion itself remained integrated in the business and the market leader). The primary intent was to benefit from the marketing know-how and resources of outside device suppliers to compete with the rocketing Palm Pilot in North America. From the early 1990s, Psion had developed retail channels in the US, but had yet to achieve significant sales there. Therefore, in March 1996 Psion publicly announced their intent to open device building to license outside suppliers for their soon-to-be-launched (fourth generation) 32-bit platform, EPOC32.\textsuperscript{55} At the same time the company was also divided into hardware and software platform development subsidiaries, in the interest of better supporting the platform business with fewer conflicts of interest.\textsuperscript{56}

The opening to outside suppliers produced considerable growth in the number of products available in the EPOC system. Where Psion had in the early 1990s released new consumer devices roughly every two years, multiple products were developed each year. Psion's own development cycle dropped considerably with roughly twice as many devices released at the end of the decade in relation to the start of the decade. The devices launched by other firms also brought greater variety. For example, Philips and Ericsson each released EPOC products with networking and telecommunications capabilities; Geofox released a tiny sub-notebook with then


\textsuperscript{53}The OS, together with agenda, database, word processor, spell checker/thesaurus, world time and spreadsheet in EPOC32 required only a 4 MB ROM. Anonymous (1997). "Introducing EPOC32." Handheld Systems (July/August).


\textsuperscript{56}More broadly, the new platform was positioned by Psion as an "open standard" on the basis of continuing attempts to promote third-party software development, the adoption of a standard microprocessor architecture (ARM) and other standard communications standards (TCP/IP, Java, compact flash memory slot, etc). Tentative experiments in licensing began soon thereafter and the opening of the platform eventually became a central fixture of attempts to promote its market acceptance. Wilson, R. (1996). Psion, Partners Open Palmtop Architecture. Electronic Engineering Times.
novel touchpad; Oregon Scientific released an EPOC-based Palmtop with fewer applications and system resources that was manufactured in China that retailed for roughly half the price of the Psion Series 5 product of the time. Given the primary objective in opening the system was to gain marketing support in the US, and given the limited development support offered by Psion in the late 1990s, the novelty and increased selection brought by outside suppliers was notable.

The intended outcome of stimulating adoption of EPOC devices in North America was not nearly as successful. In what was viewed as the most important relationship they would embark upon, Psion entered into a direct resale agreement with leading MP3-player developer, Diamond, in 1999. The deal involved fully-built Psion Series 5MX devices, re-branded as “Diamond Mako.” This was a purely commercial and marketing relationship in which Diamond bought fixed volumes of devices in exchange for exclusive rights to consumer retail distribution of the 5MX in North America. Unable to sell product, this relationship quickly ended in threats of counter-suits, and accusations that Diamond attempted to “dump” devices at low prices in European markets. This situation mirrored the overall lack of commercial success of licensees. The cumulative sales of Psion’s five device supplier partners (Geofox, Philips, Diamond, Ericsson, and Oregon Scientific) amounted to fewer than 60 thousand units. Worldwide market share for all EPOC devices (mostly Psion devices) began to collapse in this period and Psion’s primary objective of kindling demand in the US was never achieved. As the market position of the Psion platform continued to deteriorate, commitments from licensees weakened and Psion also failed to attract additional partners. For example, alleged talks with larger handheld manufacturers, Sharp and Hewlett Packard, to adopt the EPOC system bore no fruit.

With a continuing loss of momentum in the “standards battle” with Palm (and increasingly also with Microsoft) in June 1998 Psion took the radical step of devolving ownership of the platform, itself, allowing outside hardware suppliers to take large stakes in the platform. Psion gave up over half its ownership in platform assets, forming the joint venture, Symbian, with several marquee mobile phone manufacturers (Nokia and Ericsson, and Motorola soon thereafter) who also took licenses to use the platform. The adoption of Symbian by leading hardware builders and associated capital infusions led the staff and development team working

---

on EPOC to quickly double from 150 employees (80 developers) to over 400 by 2001. With "a half-billion pounds and several years work" the EPOC platform would be augmented to support wireless communications functions and be able to serve as a smartphone platform. With the passage of time (largely from awaiting the deployment of third generation packet-data cellular networks), increased capital calls, Psion's disappointments in developing its own smartphones, and Psion's diminished position in mobile computing more generally, Psion's role in Symbian gradually diminished. In February 2004, Psion announced it would divest its share of Symbian. In its attempt to promote the adoption of the platform to both outside suppliers and end-users, Psion lost control.

2.4.4 Shared Control over the Platform

In contrast to the once-sprawling control of Psion, Symbian's scope of control quickly narrowed. A main reason for reduced scope of control and even the shared ownership was to create an open platform that would allow traditional mobile phone manufacturers to use a common platform and set of standards without any one party threatening to expropriate the other, but at the same time allow them to continue to offer differentiated products in much the same way they had competed in the previous decade. The growing view held by industry participants that Microsoft might "commoditize" the smartphone business (as it had the personal computer business) or that the change of technology might upset the bargaining power of phone makers added impetus to this effort. A requirement of European competition authorities in the original formation of Symbian—that licenses be made available on non-discriminatory terms and there not be a concentration in ownership in Symbian—may have in fact also helped in the establishment of this cooperation.

61 In-person conversation with Peter Bancroft, Vice President of Market Communications at Symbian (2004).
62 Psion worked with Motorola to co-develop the "Odin" communicator, a smartphone version of Psion palm-tops, but this project was cancelled after 12 months of development.
63 This became especially clear when the possibility of Nokia acquiring Psion's shares and with it control of Symbian brought on controversy before an alternative approach involving the acquisition of shares by the minority shareholders was devised. (2003). "Nokia has Psion in the Palm of its Hands." Sunday Business.
The upshot of this strategy was a relatively disintegrated scope of the platform. In addition to no longer supplying devices, the scope was reduced to no longer include network services or peripherals, as Psion had done. More strikingly, Symbian even opened the graphical user interface—a component that had long been understood as part of the core mobile computing platform—allowing outside suppliers such as Nokia and i-mode to develop their own interfaces. Opening the graphical user interface also implied that application development might be anchored to this higher layer, rather than the lower layer of the operating system, itself. In addition to opening application development through this channel, application development was also opened by porting a Java runtime machine to Symbian, which would allow developers to develop in Java rather than in a native Symbian environment. Symbian also broke with Psion tradition by allowing outside software development environments (the testing and development tools that are loaded with the libraries associated with a given platform) to be used by developers. Less emphasis was given to applications development by Symbian itself, in hopes that outside developers would become more active. Microprocessors using the now-standard ARM architecture and supplied by multiple firms continued to be used with the platform. In addition to lowered scope, pains were taken in the transformation of the Psion platform to the Symbian platform to ensure it was highly modular and configurable to better allow customization and differentiation by licensees.

This winnowed control was intended and appeared to produce more entry by outside suppliers, predominantly traditional mobile phone manufacturers. It also appeared to generate more variation of components, and particularly in devices, user interfaces and application software. However, with this greater scope for variation was also associated the potential for lost control and coordination. To counteract a possible loss of coordination that such an open system may lead to, several steps were taken. In the early years, several flexible “reference designs” (though not fully specified designs), referred to as Crystal, Quartz, and Pearl, were offered to support different classes of devices. Within each reference design, individual designs could be configured and designs were far from fully specified, creating both considerable scope for differentiation. Further, by remaining “taperedly” integrated into applications and graph-

ical user interfaces and other elements, Symbian could assure the supply of these components if outside suppliers—for whatever reason—failed to do so. The relatively concentrated shares of ownership in Symbian and associated concentrations of market power in device development also left leading firms in the system with clear incentives to strategically sponsor the system; Symbian actively promoted the system with licensing, business development and engineering support with fees in regular operations, and also drew on capital infusions from its owners to continue to fund technical developments.68

The strategy to devolve control of the platform to hardware developers, and then allowing them to compete on differentiated products while still coordinating on common development costs and compatible standards appeared to resuscitate the EPOC platform and returned the platform as a viable competitor in the industry. Largely on the basis of Nokia’s choosing to cut-over a fraction of its huge mobile phone supply to Symbian OS rather than proprietary Nokia OS systems, 14.4 million Symbian phones shipped in 2004, 33.9 million Symbian phones shipped in 200469 (an industry leading share of mobile computers by unit volume), and a growing number of leading mobile phone makers would choose to develop devices on the Symbian platform.

2.5 Palm

Patterns of vertical integration by Palm exhibited uncanny parallels with those of earlier market leader, Psion.70,71 The differences between the cases are perhaps in the extent to which Palm was more proactive in implementing the same sorts of policies and the extent to which these were taken. Just as Psion, early development and production began as highly integrated to coordinate challenging cross-component design compromises, consistent with the Transaction

---

68 Telephone interview with Andrie de Vries, Competitive Intelligence Manager at Symbian (2005).
69 Company reports.
70 The developer of the Palm System began as venture-funded Palm Computing, was then acquired by US Robotics, which was acquired by 3COM, then spun-out as an independent firm, Palm Inc., which then eventually separated to an independent platform supplier, PalmSource, Inc. I will refer to the supplier of the Palm OS platform throughout as “Palm.”
71 This section draws substantially from interviews with managers at Palm conducted with PalmSource managers in early 2005, and phone conversations with past managers from 2003 to 2005. Additional important references included, “Piloting Palm” by Butters and Pogue (2002) John Wiley and Sons. Data on the scope of the platform was compiled through exhaustive review of all 3COM, USRobotics, Palm Inc, PalmSource, and PalmOne press releases and articles written on the companies found in the Factiva database, a compilation of periodicals, during the period of study. Details of product features were drawn from the database described in Chapter 3.
2.5.1 Palm Pre-History: Failures in “Modular” Development

Palm Computing was founded in January 1992 as a mobile computer software supplier for the Tandy Zoomer,\(^7\) an early consumer-targeted 16-bit PDA. The Zoomer had been conceived within Tandy’s Grid Computing division in large part by Palm founder, Jeff Hawkins where

---

\(^7\)Alternately, the Tandy “Z-PDA” or Casio “Z-7000.”
he was the Vice President of Research. In contrast to Grid's own relatively controlled and integrated approach to developing the Gridpad pen computer—among the first successfully-launched pen computers, launched in 1989 for insurance, oil and other industrial companies—

The Zoomer project functioned as an alliance of multiple expert firms, each contributing specialized components to the project. Casio and Tandy designed hardware. Geowork's contributed its GEOS operating system, which required major modifications to accommodate use on a mobile computer. Intuit and Compuserve would contribute applications. Palm was to develop personal information management (PIM) applications, handwriting recognition and PC-synchronization software.

While the project began with a similar vision to Psion of providing a powerful, pocketable computer and even used a similar microprocessor (a 7 MHz processor with Intel 8088 core, where Psion at the time used a 8MHz 8088 core processor), the Zoomer project would fail to be a commercially viable design. The re-purposing and attempted reduction in size of an existing operating system rather creating parsimonious code from the ground-up, while adding handwriting recognition and numerous applications—all within a first iteration of the product—would slow system performance in the best of cases. The simultaneous isolated design of hardware by Casio and Tandy, operating system by GeoWorks and applications by Palm led to further attempts to continually adjust to one-another's design changes. Tenuous 6-way negotiations in monthly meetings between the partners led to slow progress, technical compromises and an expanding set of system requirements. Clashes between partners spanned the gamut of issues including disputes over priorities, pricing, joint selling and specifications. Tension was also created as firms in the alliance—Tandy and Casio and also Geoworks and Palm—were also cognizant of the possibility they would compete in hardware and software sales and development eventually.

---


74 Grid built the Gridpad around the standard Microsoft-DOS operating system, designing all hardware, software (and underlying development tools) to specialize devices for industrial use.

75 The founder of Palm, Jeff Hawkins, was also central to conceiving this product in his role of VP of Research at Grid. Barnett, S. "Jeff Hawkins: The man who almost single-handedly revived the handheld computer industry." Pen Computing(33).


These issues and clumsy arms-length coordination of technical development (with hardware, operating systems and applications designs continually attempting to adjust to one-another) ultimately led to a product characterized by "sluggishness, inconvenient size and weight, poor handwriting recognition software, and a steep price ($700)." Palm managers also felt the larger hardware partners, Tandy and Casio, failed to invest sufficiently in marketing. The Zoomer sold 20,000 units immediately after its launch late in 1993, but sales quickly petered out. After Palm began work on improvements for a second version of Zoomer they were surprised to learn Casio and Tandy would cease support for the project in 1994. Fortunately, during the denouement of Zoomer, Palm marketed its PC-synchronization kits to numerous other high-profile systems of the day, including Hewlett Packard’s 200LX and Omnigo products and the Apple Newton. Early versions of its handwriting recognition software, Graffiti, were also made to work with Apple’s Newton in the interest of pursuing additional income.

2.5.2 Integrated Early Development—Despite Lack of Experience

Believing that a fundamentally different approach needed to be taken in the industry (and benefitting from Palm manager’s decade of experience across a number of failed systems projects...
with multiple firms contributing\textsuperscript{84}), in the Spring of 1994 Palm decided to build an integrated system on its own by taking control of design and development across the system.\textsuperscript{85} Further, this would be accomplished with a particularly challenging set of design goal of delivering a tiny, pocketable, simple mobile computing device with pen interface that would respond with negligible latency and no boot-up sequence ("instant on") and would synchronize with a PC, for under $300. At the time Palm had not built an operating system in its history, nor did it have a single hardware engineer on its roughly two-dozen person staff (including 9 engineers).\textsuperscript{86} In further seeming irony, the strategy taken by Palm would have the company also arguably place least emphasis on areas where they did have world-class experience—in software applications and utilities and to do so autonomously. The company and its venture capital investors had taken this course with the belief they would gain support and capital from a strategic investor with complementary skills and resources. But talks with Motorola, Compaq and other potential strategic partners had fallen through because of a combination of differing product visions and desired specifications, concerns about Palm's control over the platform and concerns about competition with Palm. Palm CEO Donna Dubinsky is quoted as saying: "We can't do a deal if we lose control of the product. We've gone through this with Casio [in the context of the Zoomer project]."\textsuperscript{87}

In similar fashion to Psion, Palm deliberately chose to build its system around existing, mature, off-the-shelf components\textsuperscript{88} and to integrate these components tightly around one another, largely on the basis of tightly cross-component design and parsimoniously design software.\textsuperscript{89}

\textsuperscript{84}Palm founder, Jeff Hawkins, may have also gained confidence in a closed and integrated approach from experience prior to Palm. Hawkins had acted as Vice President of Research at GRiD Computing, a relatively successful early producer of "slate" or "tablet" computers, which it produced in a relatively integrated and closed manner. There, a predecessor to Palm's hand recognition software, Graffitti, call PalmPrint had been used in a tablet computer called GRIDPad 1989. Venture Capital backers of Palm also had direct experience of having invested in multiple mobile computer ventures to serve as a benchmark in this decision.


\textsuperscript{88}The design was based in large part on a low-power Motorola 16-bit 68000 chip running at 16 Mhz (slightly slower than contemporaneous processors in Psion devices). The processor came with ready-made integrated drivers for screen and other components.

\textsuperscript{89}At a presentation at MIT in 2005, Jeff Hawkins stated: "There was nothing special about the pieces of the PalmPilot, it was how they were put together that was important."
Just as in early development in Psion, the process was one of optimizing, configuring and co-specializing components. A most obvious example of this was the creation of handwriting recognition software that was designed to be read on a specific area of the screen. Perhaps less obvious, the use of Palm’s Graffiti pen interface reduced both size and power requirements of the device. Trade-offs in memory, power and processor use also pervaded the design. Software applications were installed in ROM memory alongside and intermingled with the operating system and tested to excruciating detail for latency, and even individual pixel counts were weighed for interface usability on a coarse monochrome display—all while determining placement of active touch screen, button layout and physical design and overall user interface attributes. Rather than respond to externally imposed specifications and set components of the system, compromises and trade-offs across design elements were part of the regular development process, even choosing power source (batteries) based not only on power. The company carried out this integrated innovation on its own, in a small teams of collocated hardware and software designers. The operating system and pre-installed applications took up just a 320k footprint upon completion.

Just as the early history of quantum advances at Psion, the approach to designing the system as a tightly-knit group within a single company paid-off. After launching its first product in 1996, the Pilot, sales rocketed past one million units in the 18 months, fomenting some of the fastest industry growth in the history of consumer electronics. Within a year of sales, Palm overtook Psion’s sales share with roughly half of all mobile computer sales and would itself begin to receive awards for its novel design. Once again, a small integrated system developer had taken leadership in an industry where collections of first-tier industry veterans, such as AT&T, IBM, Microsoft, and other with many more resources working with highly modular designs and multiple firms would fail.

Beyond the initial release, control over multiple components served to ensure coordination and integrated innovation in numerous instances thereafter. Of the 39 products that Palm would

---

91 Butters and Pogue (2002) describe these aspects of Palm development, centered around founder Jeff Hawkins, in some detail.
release over the next 8 years, many were incremental extensions, but quantum advances such as the wildly popular Palm V (a radical slimming of the original product within an anodized aluminum case, commercially released in 1999) and the less popular but important innovation of the Palm VII (wireless data communicator, also commercially released in 1999) would require similar system-wide redesign. The Palm VII coincided with extension of the Palm system to the Palm.net server-side platform, whose development was also tightly linked to other layers of the system benefitted from Palm’s coordinated control in terms of its technical development. (Just as Psion’s ventures to network applications, the proprietary Palm venture failed to attract many users and was closed in July 2004) Less dramatically, the roughly annual updates of the operating system would coincide with advances in products, despite the modular design of the system. For example, the operating system and utilities began to support networking in 1997 as the Palm Professional, released that year, came with 1MB of RAM memory—sufficient memory to load a communications protocol (TCP/IP) stack, thus enabling hot-synching over LAN networks as well as modem use.\footnote{Winton, G. (2001). Palm OS Network Programming. Sebastopol, CA, O'Reilley & Associates.}

2.5.3 Tapered Disintegration & Opening the System

The basic workaday applications software programs were developed by Palm itself (including daytimer, to-do list, memo writer; and later e-mail client, browser) and effectively became part of the platform in being common elements of all devices in the Palm system (and also being stored in ROM memory)—producing benefits of tight integration and assured supply of a critical part of the system. However, Palm was only tapered-integrated into software supply. Shortly after the initial integrated development of the system and launch of a first generation of products, the system began to be opened in the interest of exploiting the development of outside component suppliers and fomenting network effects.\footnote{In-person interview Larry Berkin, Senior Director of Palm (Software Developer Support) (2005).} While building the original Palm Pilot system in 1995, Palm had a mind to soon open applications software development and so designed the architecture of its system to ensure easy loading of third-party software by synchronizing with the PC desktop, by defining a library of application programming interfaces to be used by programmers (APIs), by defining “conduits” to facilitate interaction with PC
applications, by allowing programming in multiple languages (C, C++, Visual Basic, Java), and by otherwise ensuring the feasibility of using third-party software by the device itself, in the original system design and build. The basic programming framework, API library and instructions for using the PC-synchronization conduit were also made available to Metrowerks, a leading provider of development environments so it could ship development tools. Deals were also offered to programmers for combined development kits and reduced prices on devices. A licensing regime (although relatively toothless) was set up to also formally transfer tools and intellectual property, such as the source code, to the use of programmers. The view that a wider selection of software would increase demand for hardware devices. Hundreds of applications and “hacks” (add-ons to improve the functioning of the basic system without changing the underlying system) were developed in the first year and this would balloon to over 20,000 applications by 2004. Qualitative analysis (McGahan et al.) and econometric analysis (Nair et al. 2004) both suggest widening supply of software was complementary to device sales.

Device development was opened soon thereafter. However, rather than waiting until competition necessitated a wide opening of the system as Psion had done, Palm chose to gradually open hardware supply to exploit an outside network of suppliers. The process began a year after the original launch with first round of outside device suppliers limited to value-added resellers (VARs) of re-branded Palm devices, including deals with IBM and Franklin Covey who each had a captive set of customers to whom they could sell the Pilot. Value-added resale agreements to come, however, would soon involve changes to the products themselves, adding specialized hardware and software to cater to the needs of specialized markets, such as Cresenda (wireless), Epocrates (medical applications), and Supra (real estate applications). These sorts of deals created some level of product variation and appealed to heterogeneous customer groups. The logic was similar to the opening of software, in the sense that more hardware sales by partners implied more platform sales by Palm.97

96 "Codewarrior for Palm" was available just one month after the release of the product, allowing developers to code Palm programs in C, on the Macintosh desktop. A year later, a PC-based IDE would be made available, but by that time, pioneering Palm developers had already developed their own tools and development kit and made them available on the Internet. Perhaps most important among these was the Alternative Software Development Kit (ASDK) by Darrin Massena, which bundled together the tools, headers, support files, samples and documentation necessary to develop Pilot applications on a PC running Windows 95 or NT.
97 These licensing patterns were inferred from the history of product releases, recorded in the product database.
The development of newly designed products by outside suppliers began slightly later with specialized licensees producing important leaps in specialized niches by 1998, such as industrial grade devices and even early smartphones. The earliest licensee, Symbol technologies, would develop industrial grade devices with functions such as barcode readers and networking capabilities. Among the first products by outside suppliers was even a smartphone, built by Qualcomm. After initial reluctance to give up the mainstream market for devices⁹⁸ (similar to Psion's reluctance to give up its profitable hardware business), the licensing of outside device builders eventually included companies who would more directly compete with Palm, such as TRG, Handspring and Sony. This was largely in anticipation of meeting expected continuing acceleration of demand around the late 1990s⁹⁹. Often these and other licensees (such as TRG, Handspring, and Sony) would themselves need to achieve the benefits of making cross-component decisions and adjust the operating system itself, to make advances just as Palm had done. Therefore, while these arrangements continued to expand the variety of devices, the more profound innovation performed in these cases required greater cross-component coordination, which Palm enabled by allowing alterations of its operating system. For example, TRG needed to alter the system to accommodate the first implementation of compact flash card adapters in the TRG Pro product (1999). Sony required numerous operating system changes to accommodate the growing multimedia functions of its devices. Because Sony's alterations of the operating system would be more profound, involving the creation of multimedia capabilities, contracts were drawn to explicitly detail how these improvements would be eventually brought into the platform used by all licensees and Sony took at 6% share of Palm. Therefore, the treatment of outside device builders appeared to be a compromise between opening device development to expand the market for the Palm system, while maintaining coordinated development to successfully implement product innovations. The approach led 124 of the 163 devices released to year end 2004 to come from outside firms, contributing a considerable amount of variation in product types, features and advancing performance levels (Boudreau 2005).

---

⁹⁸ Even seemingly attractive deals with firms such as Nokia were turned down.
⁹⁹ In-person interview with David Nagel, former CEO of Palmsource (2005).
2.5.4 The Separation of Platform from Hardware Businesses

Also analogous to case of Psion’s sale to Symbian, the rise of competition to Palm in the mid-2000s and the pressing challenges of gaining foothold in the emerging smartphone market opportunity led Palm executives separate Palm’s device building business from the platform business. The view was the firm could better support the “Palm Economy” more effectively and with fewer conflicts of interest (and especially in relation to other hardware suppliers) if it itself was were not in the hardware business. The advancing of component technologies, such as industry-standard ARM microprocessors, designed for low power mobile use, facilitated further separation of control of design tasks. In future years, the patterns of vertical integration of Palm appear to have continued to reflect a tension between a logic of opening the platform to take advantage of contributions by outside suppliers (even porting the platform to Linux kernel, announced in 2005), while further integrating to areas of development in mobile and network services (through Palm’s acquisition by mobile computer network applications supplier, Japan’s Access Inc.).

2.6 Microsoft

Patterns in the Microsoft case differ in important ways from preceding Psion and Palm cases. To begin, much of the development history of Microsoft was unsuccessful. Microsoft diverged from the earlier systems projects in the firm’s adherence to highly modularized design and

---

100 In-person interview with David Nagel, former CEO of Palmsource (2005).
101 The reduced instruction set architecture and specialized designs of microprocessors designed with an ARM core allowed devices to advance to chip speeds beyond 200 MHz from the maximum 68MHz available with Motorola’s dragonball architecture.
102 Numerous industry analysts and managers, however, speculated that the separation of operating system and hardware development still came with design costs. PalmOne, the hardware manufacturer, considered repurchasing the operating system supplier, PalmSource, in 2005. Wildstrom, S. (2005). Palm Taps Microsoft. Business Week.
103 Microsoft managers declined to be interviewed for this research. This section draws substantially from “Inside Windows CE” by Murray (1998) Microsoft Press. Interviews with managers from other platforms, and from industry watchers, such as those at HPCFactor (www.hpcfactor.com) also served to corroborate facts and provide insights. Additional review of developer documentation provided by Microsoft also helped further discern key points. Data on the scope of the platform was compiled through exhaustive review of all Microsoft press releases and articles written on the company (which included keywords suggesting their relationship with mobile computing) found in the Factiva database, a compilation of periodicals, during the period of study. Details of product features were drawn from the database described in Chapter 3. A thorough review of secondary sources and articles was depended on to compensate for access to primary data in this case.
conformability to existing desktop architecture created constraints in fully exploiting integrated design. In cases, the company even had multiple mobile computing systems projects which competed against one another internally—a far cry from tight-knit, integrated teams in early Psion and Palm. Despite relatively high levels of integrated control across system design, design tasks were constrained and less able to benefit from tight cross-component specialization to produce efficient designs. The introduction of large specialized components, notably chips, also created some challenges. Only in deviating from these constraints was Microsoft finally able to develop a commercially viable system, ten years after entering the market. In a sense, the goal to be compatible and open to multiple systems and to allow third party hardware and software developers clashed with the goal of tight, integrated development, even in instances when the company exerted extensive property rights and design control over the system.

2.6.1 Early Autonomous Innovation on a Small, Standard Operating System

Microsoft’s commercial beginnings in mobile computing were originally an unintended outgrowth of complementary innovation in the MS-DOS desktop platform, giving Microsoft an early presence in mobile computing, second to Psion by unit sales in the early 1990s. Some of the earliest commercial product developments of mobile computers were produced by licensees of the MS-DOS operating system intended for use on desktop personal computers and associated standard components, including Intel x86 microprocessors. Devices rapidly multiplied, produced by a number of suppliers, including: Atari, Poqet, Bicom, Compaq, Data General, Fujitsu, Grid, and others. The use of DOS and DOS-compatible systems led the first wave of 16-bit system in the late 1980s and early 1990s to even be dominated (in terms of number of product releases) by Microsoft systems—despite Microsoft’s unintended role in this. The relative efficiency and still small footprint of the MS-DOS system (in terms of use of system resources)\textsuperscript{104} and ready availability of standard parts and compatible software made the system relatively attractive for development. These attractive features would lead MS-DOS mobile devices to continue to be released through the decade to follow, even after Microsoft Windows had replaced MS-DOS on the desktop. Hewlett Packard’s 95LX (1991) was among the most popular devices at the time and included DOS-compatible desktop software embedded

\textsuperscript{104} MS-DOS of the time was roughly the same size as today’s mobile Linux implementations.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Content &amp; Services</td>
<td>MICROSOFT (MS-DOS)</td>
<td>MICROSOFT (PcN for Windows, WinPad, Pulsar, etc)</td>
<td>MICROSOFT (Windows CE)</td>
<td>MICROSOFT</td>
<td>MICROSOFT</td>
</tr>
<tr>
<td>Framework Network</td>
<td>ATARI, POQET, HP, ETC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Applications</td>
<td>IDE &amp; Tools</td>
<td>Framework &amp; APIs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operating System</td>
<td>GUI</td>
<td>Utilities</td>
<td>Kernel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device &amp; Board-Level Design &amp; Integration</td>
<td>High-level definition</td>
<td>Board-level Specs or Tools</td>
<td>Full design and integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Drivers and configuration</td>
<td>Microprocessor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peripheral Hardware</td>
<td>Development of DOS-based mobile computers with standard components</td>
<td>Failed development of multiple object-oriented, modular systems</td>
<td>Initial launch of Windows CE</td>
<td>Significant upgrade of Windows CE-based platforms</td>
<td>Promotion of adoption by providing reference designs</td>
</tr>
</tbody>
</table>

Figure 2-5: Vertical Integration Trends of Microsoft (in Mobile Computing)
in ROM while retaining significant battery life, and featured small-scale keyboards, built-in
programming language and several scaled-down desktop programs. The use of standard parts
and some degree of specialization and integration of components was to some level analogous
to the early approaches of both Psion and Palm. However the integrated development to mold
design features across components was done in hardware and software around a standard oper-
ating system (DOS) and closely associated standard Intel chip architecture known to personal
computers.\

2.6.2 Failed Attempts to Adapt Existing Technologies to Ambitious Visions

Microsoft had had a pen computing development team since 1988, but witnessing growing
interest in mobile computing in the industry, Microsoft chose to more deliberately pursue the
opportunity. The vision was to create a family of modular, scalable operating system that
could accommodate different levels of functionality and resources for collaborating platforms
built on a common code base. These would be used in a variety of devices including handheld
computers, mobile telephones, automobiles, office machines, household appliances, home enter-
tainment devices and others. The intent was to create enormous economies on a consolidated
Microsoft platform across multiple applications, allowing the company to become dominant
across a range of applications.

Early related projects resulted in the launch of a “run anywhere” embedded OS, in 1993,
“Windows for Pen Computing”—essentially a set of pen extensions, on top of a pared-down
Win16 kernel, which itself was built on top of MS-DOS—was released as a graphical user
interface operating system for mobile computers. The appeal of this platform was as an aug-
mentation and advancement on the MS-DOS platform in mobile computers that had already
shown some success. Several products were released but the ponderous code base overwhelmed
the advantages it might have had over the simpler MS-DOS OS. Handwriting recognition tech-
nology at the time was also of limited quality.

105 These early DOS systems also tended not to become burdened with pen interface or wireless capabilities,
which may have further burdened the off-the-shelf DOS/x86 chip architecture.
109 Microsoft instructed developers to limit the amount of handwriting recognition actually designed into the...
Shortly thereafter, seeking to promote the adoption of the operating system, Microsoft joined Intel and several OEM manufacturers to build a new device around this operating system intended to compete with the Apple Newton, which was receiving a good deal of press attention at the time. This was to be the “WinPad,” a 16-bit mobile device that was intended to access and synchronize data on the PC desktop while providing mobility of use, and to be able to communicate wirelessly between WinPads. Microsoft retained control over the architecture, applications and software, although it acted in consultation with Intel (which contributed its 386 chip) and OEM partners who would build the device, once specified by Microsoft. But in the end the operating system itself simply consumed too many resources to integrate effectively in a mobile device; using a subset of the PC code base with modular extensions rather than tailoring a specific operating system to the task proved itself to be infeasible.

The design changed to then use the leaner Microsoft at Work, a modular embedded OS designed for use in office machines like copiers and fax machines. And rather than use the 386 chip, Intel then set about to build a specialized microprocessor (“Polar”) in partnership with VLSI, to better meet the processing, power and price requirements of the mobile environment. This new version of WinPad then become delayed by two years, system requirements ballooned, estimated product costs more than doubled, Intel became more focused on the booming PC market, and device manufacturers began to leave the project.

In the meantime, Microsoft also began to work on a second mobile computer project to develop a small mobile computing device that would communicate through the paging network, called “Pulsar.” It was intended to leap ahead to exploit the new 32-bit Windows NT and Windows 95 object-oriented operating system framework and code base. This project immediately encountered problems in adapting the desktop kernel or its object oriented approach to the Pulsar application. The development team also struggled to retain the programming interaction with end-users.
interfaces (APIs) that would allow Pulsar would remain familiar to programmers with experience in Windows 32-bit development. Exacerbating these challenges, the operating system was at the same time intended to be used in interactive television set-top boxes. Progress in the project slowed as the goals of the mobile computer and set-top boxes diverged, and the project became much delayed. In 1995 this latest project folded.

2.6.3 Integrated Development to Produce a Highly (Overly) Modularized System

In 1995—now 6 years after the original appearance of Microsoft mobile computing systems—the WinPad and Pulsar resources and personnel were consolidated. The project was to take the 32-bit RISC architecture of Pulsar and to build on it and operating system accommodating the Windows NT kernel and object-oriented framework. The hope remained that advancing component technologies would eventually support the strategy of a common code base.

However, a new code base was finally decided upon to be developed based on a newly built kernel, called Windows CE, in order to accommodate still severe system constraints (and memory use, in particular). The multi-tasking operating system retained the desired scalability and even allowed the use of the familiar Windows API libraries (although the code was itself needed to sacrifice direct compatibility with the Windows code base to feasibly achieve these other goals). The code-base was exceptionally modular and formed of individual components—even more so than Windows 95 or 98—as was originally envisioned. With the modular operating system code base, Windows CE, it possible to configure tailored platforms to accord with the applications in mind by selecting just those components required to serve the use. Consistent

---

115 There were even tensions over which API set to use, the well-known Win32 library or those based on the new object-oriented Windows 95 kernel. The development team successfully argued for the Win32 interfaces.
116 The television application was to be a simpler, closed system dedicated to television use and with high data transmission and networking requirements, whereas the Pulsar was to support open development by third-party software suppliers. Murray, J. (1998). Inside Windows CE. Redmond, WA, Microsoft Press.
117 "We believed that Moore's Law would continue to function and that this hardware platform would become economical as we popularized Windows CE for the next generation of intelligent appliances." Kotiwicz
118 "We started with a pretty advanced object-based operating system, but when we started looking at what it took to build applications and system components, we realized it was an uphill battle. [W]e... started porting components from Win32. We ended up with probably 80 to 90 percent of Windows, but it was big and didn't give us the flexibility we needed. Approach number three was to rpece the API model but not the code." Harel Kodesh of Microsoft quoted in Murray, J. (1998). Inside Windows CE. Redmond, WA, Microsoft Press.
119 However, in the first release of Windows 1.0, tools to develop custom configurations of the platform were not
with Microsoft’s longtime vision of a highly modular object-oriented operating system, beyond mobile computers the Windows CE code base to in principle serve an extremely fragmented set of opportunities in embedded computing such as industrial controllers, set-top boxes, video game players, on-board automobile controllers, and other possible custom applications—anything with containing a processor, memory and internal clock. The ability of the Microsoft system to work with multiple microprocessor architectures further increased the potential of the system to a wide set of applications. Not only did this differ from Palm and Psion’s approach in its general applicability but also in partly competing with existing internal projects, such as Windows NT Embedded.

The particular platform shipped to OEM hardware partners for mobile computers by Microsoft was one such platform that had been configured and optimized by Microsoft (and accompanied with ready-made device drivers), the “Handheld PC” (HHPC) platform, launched in autumn 1996. In this fashion, subsequent releases of the operating system came with a slowly expanding set of pre-configured platforms. (The second platform release in 1998 came in five configurations, such as Auto-PC, an embedded system for automobiles, and Palm-sized PC, a pen-pad PDA, similar to the Palm Pilot.). This configuration of several platforms was analogous to Symbian’s creation of multiple platforms to achieve greater product variety.

Thus, unlike Psion and Palm who had exploited a highly closed, integrated and controlled initial development process to focus on product design per se, the highly controlled development at Microsoft was geared to facilitating the development of complementary products: hardware developers could use multiple configurations of the platform or develop one themselves; software developers would be able to use existing programming frameworks and tools; rather than a single standard processor architecture to work with (as was the case with Psion and Palm), any of five chip architectures could be used.

The additional design constraints and highly modular code used to achieve these ends, however, imposed severe performance costs and ironically limited flexibility of design. Along with specially-configured platforms were associated hardware “reference specifications” that

yet available from Microsoft. In coming years, Microsoft would begin shipping to hardware developers the tools that Microsoft was using in-house.

120 "I hope to have thousands of OEMs building Windows CE devices over the next few years." Microsoft manager, Frank Fite, quoted in Murray, J. (1998). Inside Windows CE. Redmond, WA, Microsoft Press.
provide recommended hardware and software configurations to achieve optimal performance. Apart from significantly degraded performance that deviation from these prescriptions entailed, it was simply not feasible for Microsoft engineers to test and validate each possible configuration of the platform and hardware and software. This may in part explain the little variation between HHPC 1.0 devices released in 1996 and 1997. Thus there was very limited variety in the first devices, with the six OEM manufacturers—drawn largely from the ranks of PC OEM partners—releasing grey or black large folding palmtops with keyboards (slightly larger than those built by Psion) with similar functions and aesthetics.

Even in conforming to the optimized HHPC design, performance was sub-par; critics in 1996 gave stinging reviews for power consumption, slow responsiveness and complexity of the interface (which had been designed after the personal computer desktop) of the early product releases. Given the challenge of developing successful mobile computing system at the time, meeting the constraints of highly modular design, an API set that included a great deal of redundancy, the multi-tasking ability of the operating system, while porting to multiple processors, and including mobile versions of Microsoft’s word processor, spreadsheet and browser software, along with PIM functions, appeared to place several too many constraints on the design.

Customer interest quickly waned. And this remained the case with the second generation platform released in 1998 with even more microprocessors supported and many more device builders creating new mobile computers—with slightly more variation in products. Even though these products now included those in a penpad form factor, similar to Palm’s successful products, the products were judged to suffer from the very same fundamental problems of the first generation: slow performance, poor battery life. The device was also criticized for continuing to appear too much like the desktop PC. Poor interface and usability. The handwriting recognition component that had been added to the system also generally disappointed. Mobile computers based on Windows CE remained far behind Palm in the market, capturing roughly a tenth of all sales while Palm captured more than half in the late 1990s.

122 The most novel of these products were those that developed custom code base using the Microsofts platform builder tools.
### 2.6.4 Increasing Integration, Tightening Specifications and Increasingly Successful Products

With its third version of Windows CE, Microsoft adjusted its design approach. This included a ground-up redesign of the user interface as the requirement of conforming to the PC desktop was dropped. The interface was rebuilt to more closely emulate features of market-leader Palm, removing a number of features to the Windows desktop. The platform was no longer ported to a variety of architectures, and ported to just one de facto standard architecture (ARM) supplied by multiple vendors. Perhaps most crucially, the reach of both hardware and software specifications was further extended and tightly defined so as to further optimize the Windows CE 3.0 code to what Microsoft expected to be the commonly encountered uses of Pocket PC products and to better assure compatibility with applications software, development environments and middleware. Additionally, a greater number of core applications software titles came pre-installed on the devices, such as the media players and e-mail client and utilities and drivers for supporting numerous components from cameras to networking became increasingly integrated into the platform. These changes would lead to less flexibility in design and less variation in products and even fewer handheld computer releases as large computer makers (such as Hewlett Packard and Toshiba) began to dominate the hardware business on the basis of brand, distribution and economies of scale and small firms from the late 1990s (such as Everex and Novatel) were less able to successfully compete with less differentiation possible. While it remained possible for small firms to experiment with smartphone models for some time, adapting the Pocket PC to these applications (such as Cesscomm and Cyberbank), eventually Microsoft would enter into the provision of standard hardware reference designs in this area and similarly eliminate the scope for small scale experimentation. Cases where new

---

124 "Although... Pocket PC is Windows CE 3.0 under the hood, the differences between Pocket PC and prior versions of Windows CE are dramatic. Windows CE was put through the wringer by many users and deservedly received a great deal of criticism... For these reasons and more, Microsoft set out to develop a completely new handheld operating system when it went back to the drawing board for Pocket PC." Morrison, M. (2001). Pocket PC: The Unauthorized Guide. Indianapolis, Indiana, Que.


127 Firms opting out of the standardized design could still use Microsoft’s “Platform Builder” software to configure Windows CE software to their needs.

firms were able to enter tended to be isolated to low-cost Asian manufacturers with an ability to economically manufacture models based on the reference designs. With this approach, the quality of Microsoft products rose markedly and Microsoft devices would steadily expand its market share with radically improved product designs and would come to parity with Palm devices by 2004. Despite the reduced number and variety of both hardware makers and chip architectures resulting from its increased integration into core software and hardware design, the systems remained open to third-party software developers (the reduction of variation in hardware and microprocessors, along with deep integration into development tools, in fact facilitated software development). Thus the tension benefits of the integration with opening to stimulate innovation of complementary components was negotiated by integrating further into some components while keeping others open.130

2.7 MontaVista’s Mobilinux

Although Linux has a considerably shorter history in mobile computing (commercial products launches only began in the very late 1990s), mobile Linux-based platforms are given attention here both because of their rapid adoption by leading device manufacturers and microprocessor manufacturers and because of a what might seem to be a distinct approach to organizing production.131 The following discussion focuses on the current leader among mobile Linux distribution providers, MontaVista, supplier of Mobilinux.132

Despite the distinct open source model in the development of the Linux kernel (that MontaVista essentially customizes and tailors to embedded and mobile applications), the essential problem of developing a commercially viable system as a whole across multiple components

130 At the 2005 Microsoft Mobile and Developer conference, Bill Gates reinforced this point: "If you look at the history of Microsoft’s successes, going all way back to the original BASIC or MS thing they all had in common is that we succeeded because we reached out to developers," Gates said during a keynote address.
131 This section draws substantially from interviews with the CEO of Montavista and other suppliers of mobile Linux platforms. Data on the scope of the platform was compiled through exhaustive review of all Microsoft press releases and articles written on the company found in the Factiva database, a compilation of periodicals, during the period of study. Details of product features were drawn from the database described in Chapter 3.
132 Recent Montavista product releases comprise roughly two-thirds of all recently-launched Linux-based smartphones, based on data collection by the author. In the broader embedded computing market, had roughly half of the entire market for embedded Linux distributions in 2004 (Fuji-Keizai Group 2004). Other embedded Linux implementations included Amirix, Coollinux, K Linux , Lineo Embedix, Lynuxworks BlueCat, RedBlue Linux and Red Hat Embedded Linux.
(which each may more may not be open source) remained the same.\textsuperscript{133} Entering the market much later than market leaders Palm and Microsoft, the overriding preoccupation of MontaVista has been to promote adoption by end-users and by outside suppliers of its system. Given the relative modularity of the technology, it has been possible to pursue a highly open and disintegrated strategy so as to accomplish this. But to then retain some ability to coordinate and govern the system a cross-firm institutional body, Mobilinux Open Forum, needed to be created.

\textbf{2.7.1 Serving Disparate Applications by Porting to Multiple Board-Level Platforms}

In contrast to traditional leading platforms in mobile computing, the scope of MontaVista's "Hard Hat" Linux operating system in launch in 1999 was exceedingly narrow. The company was focused on a miniaturizing and adapting the Linux kernel to work with embedded computer systems, including industrial machine controllers and sensors, automobile electronic systems, consumer electronics and communications systems (applications that Microsoft had addressed as part of its larger Windows CE strategy with modular code base). Linux was not as light, agile, robust and amenable to embedded applications as traditional embedded operating system, such as VxWorks, itron, or a great many proprietary operating systems that have traditionally come pre-loaded on microcontrollers. But as these systems increasingly required greater computation and communications capabilities to interoperate with other systems and segments in consumer electronics, media players and mobile computers expanded, there was an opportunity for a low-cost, scalable and powerful operating system with richer utilities. Further, the proliferation of proprietary systems in the industry left potential economies of standardization unexploited. The modularity, expandability, unrestricted access to standard source code, standard APIs and lack of royalties of Linux positioned the operating system to serve this market. MontaVista

\textsuperscript{133} The incorporation of open source components within a broader system that may simultaneously benefit from proprietary components and some measure of control exerted over the system to promote coordination is not unusual. For example, Palm draws on open source development by now porting the higher levels of its platform to a Linux kernel, as announced in early 2005. Symbian has converted its longtime development language, OPL, into an open source product. Many mobile computing platforms now utilize open source development tools such as Eclipse.
Figure 2-6: Vertical Integration Trends of Montavista
founder Jim Ready, a 25-year veteran of the embedded computer industry, sought to exploit this opportunity in forming MontaVista.

In its initial development, MontaVista's development task objective was to transform the Linux operating system to meet the requirements of "embedded" computing, a distinct class of typically small-scale computing systems requiring critical control loops and mission-critical real-time responses. This implied transforming the existing Linux kernel, which required roughly 150 MB of disk space, and tailoring it to be able to run with resources as meager as a simple input-output board with limited on-chip memory (and be able to boot without access to keyboard, monitor or disk access)—while bringing quantum leaps in latency (response time) and power management. MontaVista’s team, including both Linux and traditional embedded computing designers, went through the Linux code and removed components that were not relevant or strictly necessary, while attempting to preserve as many of the essential features of Linux as possible so as to benefit from compatibility, complementary developments and continuing development related to wider development related to Linux.

The initial product, launched in 1999, “Hardhat Linux” was 11.8 MB, barely larger than MS-DOS and could even fit on a single floppy disk, of which a subset could be used. The kernel was originally built as a layer “on top” of an underlying traditional real time kernel so as to meet low latency and interrupt management requirements. Although this solution did not offer the same level a latency as in simpler, proprietary systems, The solution provided a 30-fold improvement in responsiveness on the standard Linux operating system. Shortly thereafter, in 2000, MontaVista would be the first firm to do away with the dual-kernel approach, preserving the Linux programming model and leading to an additional 3-fold improvement in operating system response time. “The charm here is that it’s standard Linux. It doesn’t change the APIs. It’s not a separate kernel. We’re following our company’s philosophy, which is to improve

---

134 Hunter & Ready, Inc. co-founded by ready developed the VRTX embedded kernel, which runs systems of the Hubble Space Telescope.
136 In July 2000, Montavista introduced a real-time scheduler for Linux that eliminated the need for a second kernel operating beneath Linux. Making use of standard features that are already a part of Linux created greater compatibility with Linux APIs. Keller, J. (2000). “MontaVista unveils real-time scheduler for Linux.” Military & Aerospace Electronics 11(7).
Linux,” Jim Ready was quoted as saying. In following years the kernel and supporting tools would continue to be improved, decreasing size, latency, superior functionality (notably a power management utility) would be added. The basic code base would eventually be expanded to several broad classes of applications—industrial embedded systems, consumer electronics products mobile phones, and carrier grade applications for telecommunications products) with a couple dozen different configurations to begin design with by 2006.

In the case of highly customized and varied embedded systems development, narrow scope was made possible by selling to embedded systems “board developers” (such as Force Computers, Ziatech, WinSystems, SBS Technologies, and a great many others) who themselves created basic hardware reference designs, including very basic board components with microprocessors integrated. In this sense, Hard Hat Linux was a component within a larger platform, taking the place of traditionally simpler operating systems that had come bundled in these board-platforms. In this, the extraordinary configurability and modularity of the Linux (MacCormack et al. 2005) system made it economical to port the operating system to many different hardware platforms.

And despite the numerous platforms of which Hard Hat Linux was a part, the tailoring of Linux tools by MontaVista and tools available for native coding on the processors created numerous options for embedded systems developers. In a few years, this approach of providing a modular, configurable, high-performance embedded system would bring the company hundreds of large customers, using 32 different microprocessor manufacturers by 2006 (many more than even Microsoft).

The applications of the highly configurable operating system would expand to set-top boxes for televisions, toy robots and even carrier grade telecommunications applications. In addition to the continual augmentation of Hard Hat capabilities, the porting to new platforms and processors led to new releases on a roughly monthly basis. And thus despite MontaVista’s narrow scope, it was able to serve a highly disparate set of applications by maintaining a highly compact, modular code base (akin to Microsoft’s original vision) and to then serve as a

---

139 Company website.
141 Company website.
component within larger platform where additional specialization to the application took place.

2.7.2 A Mobile Computing Platform Created with a Coalition of Specialized Partners

While the strategy of supporting other firm's platforms led MontaVista to move towards supporting mobile computers through its support of other platforms\(^\text{143}\), MontaVista took a special interest in developing a platform in which its Linux implementation would be the central and defining element\(^\text{144}\) in support of mobile computers—and smartphones, in particular. CEO Jim Ready stated: "This 'platform' idea that could develop around smartphones could be the battle royale. From an overall business standpoint and semiconductor standpoint, you gotta win it. It's the kind of battle that tends to bring in a lot of resources."\(^\text{145}\)

In this, the primary development activities of MontaVista remained the continuing specialization and innovation of the kernel\(^\text{146}\) itself, along with supporting utilities and tools. Montavista then simultaneously maneuvered to entice outside suppliers to supply other components that would then be used as an integral platform by outside device developers. Given tight competition between numerous embedded Linux suppliers and tighter competition with incumbent mobile computer platform suppliers (as those discussed), this strategy would accommodate MontaVista's need to rapidly extend the scope of its platform with world-class components (which it, itself, with its deep but narrow expertise did not have the experience or resources\(^\text{147}\) to develop) while avoiding a concentration of control over the platform which might discourage the platform's adoption. Thus, the development of external relationships with those firms able to provide the other essential elements of a platform to enable rapid development

\(^{143}\)For example, in 2000 Montavista's Linux kernel was ported to Instrinsyc Software's "Cerfboard" (hardware and software reference design) platform, intended to support handheld computers, mobile and Internet devices.

\(^{144}\)The earlier use of Hard Hat Linux was as a platform in the sense of a component used in common across multiple implementations, but future versions would begin to define the systems, defining architecture and defining key interfaces.


\(^{146}\)In addition to the general advance of MontaVista's platforms, variants of the platform proliferated. With more variations, each variant tended to be come more specialized to a particular application and require less customization off-the-shelf. For example, mobile computing applications once supported by Hard Hat, were later supported by the "Consumer Electronics" version, and later by "Mobilinux."

\(^{147}\)Even by 2004, Montavista staff included 60 development engineers devoted to kernel and tools development and to supporting customers.
of smartphones became a key and defining element of MontaVista’s strategy.\textsuperscript{148} This included developers of essential applications software, graphical user interface, applications development environments, microprocessors and hardware architecture reference designs.

The earliest extensions of the MontaVista platform, in relation to readying the platform for mobile computer use and smartphones in particular, related to applications development. These represented substantial extensions from an original product that did not even necessitate the presence of a display screen. As early as 2000, basic applications were ported to the MontaVista platform. For example, in 2000, the viewML browser (an open source project) was ported to the MontaVista kernel by Century Software.

More importantly, entire applications development, run-time environments and graphical user interface systems were soon added to run “on top” of the kernel, thus serving as both components within the system and also as middleware that allowed software development to proceed somewhat independent of the specifics of Montavista’s design. Java support in the form of runtime engines integrated in the operating system and in development tools and support came from Insignia (2000), IBM (2000) and Esmertec (2005). Graphics tools, graphical user interfaces (and associated application development environments) were also ported from Trolltech (2001) and Openwave (2004), which each also brought world-class user interfaces, development environments and libraries of existing software applications. MontaVista, itself, also supplied its own “Hardhat Graphics” environment (2001), offering graphics development across a range of applications. Thus, the strategy was somewhat analogous to Symbian’s opening the graphical user interface—although perhaps MontaVista benefitted even more from the differential experience of outside suppliers.\textsuperscript{149}

Especially critical to MontaVista’s in promoting outside suppliers’ adoption of the platform was in enticing chipmakers to support the platform and integrate it within their increasingly-integrated system-on-chip designs.\textsuperscript{150,151} While in many regards, system design on the basis

\textsuperscript{148} Telephone interview, Jim Ready, CEO of Montavista (2005).
\textsuperscript{149} Symbian already had experience in developing user interfaces.
\textsuperscript{150} Telephone interview with Jim Ready, CEO of MontaVista (2005).
\textsuperscript{151} Logic processing and radio communications processing, along with a growing number of key component drivers tended to be integrated on the same silicon to improve performance and power consumption. By the mid-2000s these system-on-chips tended to increasingly accompany reference designs, or direction for building working smartphones. These designs were specialized both to the operating system and chip elements of the platform.
of Linux and modern component technologies had successfully become more modular, support from and compatibility with chipmakers had become particularly crucial—and more so in the development of mobile phones in relation to other embedded applications. In addition to tightly configuring board-level and system design to microprocessors, the microprocessors themselves now increasingly incorporate both central processing functions, radio signal processing and telecommunications utilities, as well as many component and input-output drivers, leading these products to increasingly be referred to as “system-on-chip” microprocessors. Further, by the early and mid-2000s these system-on-chips tended to increasingly accompany reference designs, or direction for building working smartphones which dramatically widened the range of firms able to successfully develop mobile computers. Embedded Linux from MontaVista was therefore made available from a range of leading microprocessor manufacturers, embedded in the chips themselves, for mobile computing applications, including Toshiba, Texas Instruments, Intel and IBM, and based on industry standard (ARM) architecture. Montavista’s early ability to persuade Intel to support Montavista’s implementation of Linux and even take an equity stake in the company in 2000 may have been a critical turning point in creating this momentum.\textsuperscript{152}

\textbf{2.7.3 Creating Forbearance and Coordination Among Interdependent Suppliers}

While diffuse control and power in the MontaVista system may have been necessary to enlist outside firm’s support and to promote adoption and to bring together world-class components, the thin scope of control of MontaVista and co-specializing investments ran counter to the lessons of Psion, Palm and later Microsoft who chose to integrate precisely so as to coordinate across multiple such interdependent components.\textsuperscript{153} Higher modularity of design and growing component power by this time may have mitigated the severity of design interactions. And yet, MontaVista has been able to take an early lead among Linux developers and present a plausible threat to market leader.

\textsuperscript{152} Telephone interview with Jim Ready, CEO of MontaVista (2005).

\textsuperscript{153} While market leaders in the industry had uniformly had widely integrated platform suppliers, those projects with highly interdependent partners—such as projects involving Go Corporation, General Magic and the Zoomer project—ended in ill-coordinated development and ill-willed partners (even apart from inefficiencies of over-modularization of designs).
This may be in part explained by the extraordinary modularity of the system, which reduced interactions among designs and the scope for coordination problems. The multi-sourcing of component suppliers (and investments of chipmakers in Montavista) would have also likely reduced the ability and interest of partners to expropriate one-another. The need to avoid any sort of bad reputation in this respect while attempting to promote the adoption of the system would have also reduced the temptation to do so. In the case of the component in which there was not multi-sourcing—Montavista’s operating system—the somewhat narrow intellectual property rights over the operating system (related to special functions within the kernel and in development tools) and the associated ability to substitute MontaVista’s Linux with a different implementation reinforced this situation.

Nonetheless coordination problems were inherent in the project, as explained by Jim Ready, CEO of Montavista: “In all the other cases, the owner of the platform has complete control. The game that Linux changes so completely is in the control of the platform—the licensing, how the technology evolves, the whole bit—and nobody controls it and everybody controls it.”

In this regard, MontaVista as developer of the operating system was able to act in a special, focal coordinating role to gain coordination of parties in the development of standards. Initially this was in the form of bilateral negotiations with each of the trade partners mentioned earlier. However, as the set of partners grew, the “Mobilinux Open Framework,” (named after the variant of Montavista’s Linux specialized to mobile handset design) was formed in 2005 as an open institution intended to serve as a forum for developing and disseminating standards and handset reference architectures, in particular, using MontaVista’s operating system.

2.8 Discussion

A review of the earlier developed hypotheses and the evidence in relation to these suggests platform boundaries were highly pliant and responsive to economic incentives described by each of the different perspectives. However, the role of Power and Incentive Systems, Transaction Cost Economics and Open Systems perspectives appear to be most salient in explaining the observed patterns.

Telephone interview with Jim Ready, CEO of MontaVista (2005).
Platform Boundaries and the “Power” to Regulate Other Suppliers  It was hypothesized that the boundaries of platform suppliers would encapsulate critical components that confer the ability to regulate entry or otherwise directly set the “rules” of conduct in the system. Indeed, despite much variation in platform boundaries, there appeared to be an enduring impulse to integrate the operating system, graphical user interface, application programming framework, and high level system specifications. These were common denominators of components used in common, the technical platform. The platform components around which ownership and control were focused were critical in the sense of being strictly complementary to all other economic activity in the system: doing business in the system required doing business with the platform supplier. From an industrial organization perspective, this strict complementarity implied monopoly control over this stage of production, with bilateral relationships with most every firm producing in the system associated with the platform. The monopoly supply of critical platform assets (and associated high bargaining power) and bilateral relations with most every supplier in the system, in turn, allowed the platform suppliers—in principle—to appropriate surplus in the system, or at least have incentives to promote, subsidize and sponsor the competitive success of the system as a whole. The attempts of most every platform supplier to promote the entry and active development of application software developers is perhaps a most conspicuous manifestation of this.

In cases where platform boundaries were repeatedly and markedly decreased (Palm, Psion and Symbian), it was these components that were the last to be disintegrated or where exclusive control was given up. Indeed, it was only under the most extreme pressures to disintegrate these components, such as in the precipitous fall of market share of Psion, the loss of momentum and decline of market share of Palm in the mid-2000s or in attempts by MontaVista to compete successfully as a new platform supplier, that these components were disintegrated. In cases where platform owners began with narrower scope (Microsoft), these were among the first elements to be included. In cases where the initial platform was too narrow to include these components, they were among the first to which the platform would expand (MontaVista’s Mobilinux).

By integrating and maintaining exclusive control over critical, enabling, and system-defining assets, it appears the original innovators of these systems were able to serve as an organizing
force and coordinator across production in the entire system, despite only directly controlling a fraction of the system. These were also the components embodying most “rules” of design and the components to which outside suppliers needed to gain access if they were to be able to enter into the production of other components in a system.

Platform Boundaries and Solving Coordination Problems  The evidence supporting the setting of platform boundaries in accordance with the Transaction Cost Economic perspective also appears relatively prominently. In the earlier literature review, it was hypothesized that coordination problems would be internalized within the boundaries of a single firm. From the preceding cases, it was the platform supplier that often integrated activities susceptible to coordination failures. This was most starkly revealed in early industry development wherein the immature state of component technologies required extensive specific investments and cross-component coordination to ensure a functional product. In this context, the industry was led by a conspicuous stream of highly integrated systems, where the platform supplier was in fact also the hardware and applications designer. These systems achieved levels of performance and project completion times that far outstripped projects by (often more resource-rich) “modularized” projects that were touted as “open” standards, in which multiple expert suppliers contributed components. In another clear example of avoiding coordination problems, and hold-up problems in particular, platform suppliers integrated into core applications programs to ensure the supply of these essential components in the shipped platforms.

It was also hypothesized that the production of components used in common (the technical platform) would be carried out by a single supplier (the platform supplier). Indeed, we see that this in by and large the general situation across the history of these platforms. In addition to generally seeing this predicted pattern, we also see that when the platform itself is finally opened for other reasons, there is a general tendency to retain consolidated control of the platform under a single platform supplier (and to divide the ownership). Where there were exceptions, such as in the splitting of the MontaVista platform and the Symbian platform into multiple components, careful institutional provisions such as dual sourcing were universally implemented, presumably to overcome coordination problems the that might result. A less anomalous example of this was a relatively common case of using third party microprocessors (used in common across multiple
device builds), but being protected by the fact these components were off-the-shelf or standard components.

- “Opening” a System, Appropriability, and Interactions with Systems Competition
  It was hypothesized that components most prone to network effects would be opened. This was readily and universally observed in the early opening of applications software in the interest of stimulating large numbers of applications suppliers to enter in hopes this would stimulate a complementary response in hardware purchases by end-users. In addition to allowing entry, each of the platform suppliers would even use contests, make their own investments in supporting tools. In the case of Microsoft, the basic architectural designs even hinged on the desire to bring applications programming interfaces and modularity to the platform to facilitate development by outside applications developers.

  While these may seem on one hand to be extraordinary measures, they were far “cheaper” than costs in lost appropriability entailed in opening the lucrative hardware business, which explains Psion’s reluctance to open its hardware business, and Palm’s measured slowness to do so. In both cases, both platform suppliers retained some measure of appropriability in the hardware business through maintaining their own hardware manufacturing divisions. This may have also aided coordinated development of new products. It was only with pressing conditions of competition that “pure” platform strategies were pursued by Palm and Psion (Symbian), in the sense of disintegrating from device supply and not competing with their trade partners.

2.9 Conclusions

Are the platform boundaries responsive to economic incentives? How so? Is the economic logic analogous to that of “regular” firm boundaries? As an organizing framework to address these questions, this paper laid-out alternative explanations for how platform boundaries might plausibly be responsive to economic incentives, both relating mainstream theories of firm boundaries to those of platforms and also relating to ideas drawn from the industrial economics research that focuses on the exotic conditions of systems industries and the unique role of platforms, in particular. Data were systematically collected on the vertical boundaries of multiple platforms in the mobile computing industry over a couple decades and the logic for the placement and
shifting of platform boundaries in each of five leading platforms (and, secondarily, a handful of others) were reviewed from the lens of these theories. This archetypal software-hardware industry provided a useful context to study these issues given the coexistence of multiple platforms and the comparable structure of these platforms. Also, the industry context was one where the vertical integration decision clearly was the clear prerogative of the original system innovator (cum platform supplier), and so the patterns could be interpreted in that light.

The boundaries of these platforms were found to be extraordinarily plastic and responsive to economic incentives, with startling variation both over time and across systems (startling when considering the frequent treatment of platform boundaries as fixed and exogenous features of industry structure). From the preceding discussion, there appear to be several main factors that explain the vertical integration decisions of platform suppliers: The location and changing boundaries of platform suppliers appear to mostly be explained by three main arguments:

1. “Locating” platform boundaries so as to consolidate control around assets that conferred the power, economic incentives, means and instruments to “govern” or “regulate” the system, as a whole;

2. Extending platform boundaries to integrate economic activities that risked coordination problems;

3. “Opening” or contracting platform boundaries in response to interactions between economic organization and market competition.

Therefore, the evidence that the boundaries of the platform supplier are responsive to economic incentives would seem compelling. Theories of the firm were found to readily apply to the situation of platform boundaries. While sometimes they applied directly (as in the interest of internalizing coordination problems through vertical integration), in other issues it became clear that the platform supplier in the broader system was analogous to the firm owner to the broader firm and theories of the firm were interpreted isomorphically to a view of the system as a subeconomy, as a whole, wherein the platform supplier was the regulator. In these cases, patterns of vertical integration both reflected an interest in gathering a base of assets from which power to regulate could be derived and projected across firm boundaries, and also
to promote adoption and system-specific investments by other firms. While other key determinants of platform boundaries in this context, namely interactions with market competition, and shifting constraints and technology choices associated with technology might be understood within these existing theories of the firm, theories of the firm in the main do not yet explicitly account for nor endogenize these features.

Future projects will seek to more systematically characterize these patterns with econometric evidence, drawing principally on the quantitative database underlying this study. In there may be important opportunities to formalize the interaction of firm boundaries, market competition and technical design issues within applied theoretical models to add greater precision to our understanding of these issues.
Bibliography


2.10 Appendix
<table>
<thead>
<tr>
<th>First</th>
<th>Last</th>
<th>Company/Organization</th>
<th>Position</th>
<th>Medium</th>
<th>Date*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alessandro</td>
<td>Anzaldi</td>
<td>Texas Instruments</td>
<td>Worldwide Marketing Manager</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Peter</td>
<td>Bancroft</td>
<td>Symbian</td>
<td>VP Market Communications</td>
<td>in person, telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Larry</td>
<td>Berman</td>
<td>PalmSource</td>
<td>Worldwide Developer Marketing</td>
<td>in person, telephone</td>
<td>2005</td>
</tr>
<tr>
<td>David</td>
<td>Childers</td>
<td>Oregon Scientific</td>
<td>CEO</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Paul</td>
<td>Crossley</td>
<td>Diamond</td>
<td>Product Marketing Manager</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Michael</td>
<td>Davies</td>
<td>Industry Consultant</td>
<td>Head of Business Intelligence</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Andrie</td>
<td>DeVries</td>
<td>Symbian</td>
<td>Industry Consultant</td>
<td>in person</td>
<td>2005</td>
</tr>
<tr>
<td>Richard</td>
<td>Doherty</td>
<td>Envisioneering</td>
<td>Wireless Industry Consultant</td>
<td>Internet</td>
<td>2004</td>
</tr>
<tr>
<td>Chris</td>
<td>Dunphy</td>
<td>PalmSource</td>
<td>Product Manager, Applications &amp; Device Lifecycle</td>
<td>in person</td>
<td>2005</td>
</tr>
<tr>
<td>Jerry</td>
<td>Epplin</td>
<td>Journalist, Software Engineer</td>
<td>Stereotaxis, Linuxdevices.com</td>
<td>in person, Internet</td>
<td>2005</td>
</tr>
<tr>
<td>Jessica</td>
<td>Figueras</td>
<td>Ovum</td>
<td>Industry Consultant</td>
<td>Internet</td>
<td>2004</td>
</tr>
<tr>
<td>David</td>
<td>Gunton</td>
<td>Open Mobile Terminal Platform</td>
<td>Board Member OMTP, Senior Expert Global Products T-Mobile</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Eric</td>
<td>Gerulden</td>
<td>Sony</td>
<td>Clie Marketing Manager</td>
<td>telephone</td>
<td>2003</td>
</tr>
<tr>
<td>Eric</td>
<td>Giguerre</td>
<td>Palm, Other Software Developer</td>
<td>Author and Software Developer</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Avner</td>
<td>Goren</td>
<td>Texas Instruments</td>
<td>Director, Worldwide OMAP Products &amp; Systems Marketing, Senior Member Technical Staff</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Todd</td>
<td>Gort</td>
<td>Gartner Datasheet</td>
<td>Principal Research Analyst</td>
<td>telephone</td>
<td>2003-2004</td>
</tr>
<tr>
<td>George</td>
<td>Grey</td>
<td>Geofox, Psion, Savage</td>
<td>President Psion US, President &amp; COO Savage</td>
<td>telephone</td>
<td>2004-2005</td>
</tr>
<tr>
<td>Randy</td>
<td>Guisto</td>
<td>IDC</td>
<td>Vice President of Mobile and Desktop System Research</td>
<td>telephone</td>
<td>2002-2005</td>
</tr>
<tr>
<td>Dan</td>
<td>Hantula</td>
<td>Industry Journalist</td>
<td>Independent Consultant and Author</td>
<td>telephone</td>
<td>2003-2004</td>
</tr>
<tr>
<td>Otto</td>
<td>Hiroyuki</td>
<td>NTT DoCoMo</td>
<td>Manager</td>
<td>in person</td>
<td>2004</td>
</tr>
<tr>
<td>Virginia</td>
<td>Hodges</td>
<td>CompUSA</td>
<td>Buying Manager for Handheld Computers</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>David</td>
<td>Kipping</td>
<td>Trolltech</td>
<td>Director of Business Development</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>J</td>
<td>Kuper</td>
<td>Handango</td>
<td>Manager, Business Development</td>
<td>in person</td>
<td>2004</td>
</tr>
<tr>
<td>Timothy</td>
<td>Lehane</td>
<td>Oregon Scientific</td>
<td>Senior Product Marketing Manager</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Elisabeth</td>
<td>Liddell</td>
<td>@FoxPop (Geofox interest website)</td>
<td>Editor</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Dave</td>
<td>Limp</td>
<td>PalmSource</td>
<td>Senior Vice President – Corporate and Business Development and Chief Strategy Officer</td>
<td>in person</td>
<td>2005</td>
</tr>
<tr>
<td>Michael</td>
<td>Mace</td>
<td>PalmSource</td>
<td>Vice President of Strategic Marketing</td>
<td>in person, telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Steve</td>
<td>Mann</td>
<td>Early Palm Developer</td>
<td>Author and Software Developer</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Darrin</td>
<td>Massena</td>
<td>Early Palm Developer</td>
<td>Software Developer</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Gary</td>
<td>McFadden</td>
<td>Handango</td>
<td>Enterprise Sales</td>
<td>in person</td>
<td>2004</td>
</tr>
<tr>
<td>Kip</td>
<td>Mentzer</td>
<td>PalmSource</td>
<td>Investor Relations</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>David</td>
<td>Nagel</td>
<td>PalmSource</td>
<td>CEO</td>
<td>in person, telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Derek</td>
<td>Peachey</td>
<td>Handango</td>
<td>Senior Director of Business Development, Channels</td>
<td>in person</td>
<td>2004</td>
</tr>
<tr>
<td>Kathleen</td>
<td>Peters</td>
<td>PalmSource</td>
<td>Vice President of Sales, Americas</td>
<td>in person</td>
<td>2005</td>
</tr>
<tr>
<td>Will</td>
<td>Pinnell</td>
<td>Handango</td>
<td>Senior Director, Business Development</td>
<td>in person</td>
<td>2004</td>
</tr>
<tr>
<td>Robert</td>
<td>Quinn</td>
<td>Epocrates</td>
<td>Senior VP Engineering; CTO</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Jim</td>
<td>Ready</td>
<td>Montavista</td>
<td>CEO</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Inder</td>
<td>Singh</td>
<td>Lynuxworks</td>
<td>CEO</td>
<td>telephone</td>
<td>2005</td>
</tr>
<tr>
<td>Steve</td>
<td>Strassman</td>
<td>Orange</td>
<td>Principal Architect, Developer Lab</td>
<td>telephone</td>
<td>2004</td>
</tr>
<tr>
<td>Christopher</td>
<td>Tilley</td>
<td>HPCFactor</td>
<td>Editor</td>
<td>telephone</td>
<td>2003-2005</td>
</tr>
<tr>
<td>Jason</td>
<td>Wells</td>
<td>Handango</td>
<td>Vice President of Business Development</td>
<td>in person</td>
<td>2003-2004</td>
</tr>
</tbody>
</table>

*Many with multiple interactions

Figure 2-7: List of Interviewees
Chapter 3

The Effect of Openness on Innovation

3.1 Introduction

Innovators in systems industries (such as IT, media, consumer electronics and computers) face the question of whether to “open” their technologies, willfully giving up control and allowing outside suppliers access to enter into the development and use of a technology. While opening a technology has long been understood as a means to strategically promote adoption and network effects in industries where technical standards and compatibility play a role (Katz and Shapiro 1986), evidence and arguments presented by a wide sweep of scholars have begun to suggest that the extent to which a technology is opened also affect innovation of the technology, itself (Langlois 1992; Bresnahan and Greenstein 1999; Baldwin and Clark 2000; von Hippel 2005). The general spirit of these arguments is that opening can spur innovation by marshalling the efforts of a large and diverse pool of suppliers while stimulating virtuous circles of adoption, entry and investment. Gathering voices within popular debate surrounding open source software (and kindred debates relating to patenting,\(^1\) copyright policy,\(^2\) and technology standards\(^3\) ) even suggest the possibility that open technologies may be generally more innovative (Raymond 1999).

Impressive technological accomplishments achieved by multiple suppliers working together in open systems (von Burg 2001) and cases of open technologies out-competing closed technologies (Cusumano et al. 1992) lend credence to these ideas. However, the small body of theoretical work on the topic can be interpreted as offering a more cautious perspective (Farrell and Katz 2000; Becchetti and Paganetto 2001). For example, increased competition in an open technology system could lead to diminished investment incentives or make coordination across multiple suppliers difficult. The objective of this paper is to provide (to the best of my knowledge) a first systematic empirical analysis of the relationship between the degree of openness and innovation outcomes.

As there is yet limited evidence and general theory, the approach taken is to develop empirical hypotheses that contemplate how broad categories of factors—investment incentives, diversity of the supplier pool, and the efficacy of the coordination regime—may respond to greater openness and interact with one another. I argue that these factors should lead the sign and strength of the openness-innovation relationship to vary with two main considerations: 1) the nature of the innovation task (the extent to which the development task involves challenging coordination problems); and 2) the nature of strategic interactions among entering suppliers (whether there is competitive crowding and substitution or strategic complementarities acting between suppliers entering into the open technology).

The empirical context, mobile computing (including handheld computers and smartphones) offers a canonical hardware-software industry with the attractive features of allowing panel data analysis in a context where there were multiple competing technical systems. The extent to which original system innovators devolved control (primarily measured with a count of components under original innovators’ control) and granted access (measured by a score related to the restriction or promotion of entry) to outside device suppliers varied as a matter of degree across systems and over time, allowing the analysis of incremental changes. The devolution of control was primarily measured by counting the number of components under the exclusive control of the original innovators, as opposed to being part of the responsibilities of outside device suppl-

\[1\] The paper, however, can be interpreted as part of existing traditions of empirical research on control rights and incentives (see Lerner and Merges 1998) or technology licensing (see Arora, Fosfuri and Gambardella 2001).

\[5\] Smartphones are mobile phones also used for data and Internet communications and able to load applications software. These devices have computational capabilities comparable to handheld computers and are often built on the same technology platforms as handheld computers.
pliers. The amount of access to the platform was measured as a tabulated score that related to the general liberalism of licensing and ease of entry created by the actions and investments of the original innovators. Further, in this industry it is possible to separate multiple measures of innovation into groups of high-coordination versus low-coordination innovations; the data can also be in relation to products in the (non-networked) handheld computer submarket and the smartphone submarket to isolate conditions of low and high complementarities acting between suppliers, respectively.

Notwithstanding the virtues of this industry, the selection of empirical context should be understood as placing priority on empirical relevance over empirical convenience. Mobile computing remains a complex and fast-changing industry. So as to reveal the nature of economic relationships rather than impose a structural interpretation (and to avoid making assumptions about industry equilibrium), the approach taken is to study relationships in a reduced-form panel framework. The major identification challenge that is contended with in the analysis is the endogeneity of openness variables and is dealt with by exploiting the panel structure of the data.

The patterns found in the openness-innovation relationship across multiple measures of innovation were consistent with predictions and regular across multiple measures of innovation. Low-coordination innovations climbed sharply (≥ 30% increases with a standard deviation change in control or access measures) with increasing openness technology. Analysis of underlying mechanisms suggested that this response was the result of increasing entry with greater openness; devolving control was also found to have the additional effect of widening the scope for differentiation. The positive relationship, however, turned negative at high levels of openness in the case of handheld computers, while remaining positive in the case of the smartphone submarket. This difference could be attributed to higher competition and crowding-out of investment incentives in handheld computers where strategic complementarities were far weaker than they were in the case of smartphones. In contrast, the openness-innovation relationship in the case of high-coordination innovations was roughly zero (the frequency of innovations increased with greater openness; but the magnitudes of advances were unaffected). This was consistent with hypotheses of an ambivalent investment response to greater openness in this case, and increased difficulty of carrying out difficult coordination problems with high openness.
The paper is organized as follows. Section 2 reviews the literature and generates empirical hypotheses. Section 3 describes the empirical context. Section 4 describes the data and variable construction. Section 5 describes the empirical strategy and model. Section 6 presents results for the main regressions. Section 7 studies the nature of underlying economic mechanisms to provide corroborating evidence to support the hypotheses and interpretation of patterns. Section 8 summarizes and concludes.

3.2 Conceptual Framework & Hypotheses Development

In a most basic sense, “opening” a technology implies an innovator gives-up control and allows outside suppliers to enter (Katz and Shapiro 1986; Varian and Shapiro 1998). Intermediate degrees of openness may be implemented if the innovator can choose to devolve control over part of the technology; and in cases where the innovator retains control over at least some critical subset of the technology, the innovator retains the ability to regulate access and entry (Tirole and Rochet 2005).6

Deliberately opening one’s technology may not at first seem an obvious strategy to pursue. By allowing outside suppliers to enter into production and development, an innovator will lose monopoly over the technology (Kende 1998). Because costs in lost appropriability may be high, research and practice has tended to largely focus on opening as a mean to promote adoption7 in extreme cases of winner-take-all competition, where small differences in market share can be magnified by network effects and other sources of “positive feedback” (Arthur 1989).8

However, the radical re-organization of production implied by the opening of a technology should also influence the continuing innovation of the technology, itself—apart from any changing adoption dynamics. Given limited existing theory and systematic evidence on the openness-

---

6In polar cases of complete openness or closedness, the treatment of control and access go hand in hand (i.e. giving up property rights lead to free entry; retaining all property rights forecloses entry). In cases of partial openness, these may vary independently. The discussion will explicitly note where devolving control has different implications from granting access, but otherwise refer to both as “increasing openness.”

7Opening a technology can promote adoption and market share by reducing the threat of “lock-in” or hold-up of buyers (Shapiro and Varian 1998), by allowing multiple suppliers and buyers to coordinate on the same technology (Cusumano, et al. 1992; Augereau, et al.2005), by increasing mix-and-match benefits to users (Matutes and Regibeau 1988; Baldwin and Clark 2000), and other reasons.

8Thus the decision to open has been understood as a trade-off between pursuing a “large share of a small market” or a “small share of a large market” (Shapiro and Varian 1998).
innovation relationship, the following discussion considers three broad categories of factors that may be affected by increasing openness. The review elucidates that trade-offs between these mechanisms should make it difficult to simultaneously bring “maximum imagination and diversity to the problem while... facilitating complex coordination and strengthening incentives for product development” (Farrell and Weiser 2003).

3.2.1 Diversity

An argument figuring prominently in case study evidence (Langlois 1992; von Burg 2001) is that greater openness leads to increased diversity in the supplier pool, which in turn leads to greater production possibilities as the comparative advantages of heterogeneous suppliers are brought together. In addition to tangible differences in firm “cost curves” relating to, say, unique experience and know-how, diversity might more broadly also relate to varied motivations, beliefs and ideas (Raymond 1999; Hertel, et al. 2003), privileged knowledge of preferences (von Hippel 1998), or any other sources of differences. Diversity might also simply include differences in “luck” in searching along the technical frontier (Baldwin and Clark 2000) or in marketing to uncertain demand (Kranton and Minehart 2000). Underlying industry structure might also simply lead to endogenous differentiated product choices and Smithian specialization with greater openness and entry. Thus diversity effects, should (weakly) increase with greater openness and entry, simply on the basis of promoting greater entry. Devolving control over a greater span of the technology might also provide entering suppliers with greater scope for differentiation.

3.2.2 Investment Incentives

The changing allocation of control and amount of entry implied by increasing openness should also economic incentives to make investments in innovation. As the response of incentives of entering suppliers may not be the same as those of the original innovator, I deal with each

---

9 Appendix A provides a simple, reduced-form characterization of the openness-innovation relationship that mirrors the following explanation.

10 Research on open source software and community development processes points out that non-economic incentives can also arise around open technologies (Hertel, et al. 2003; Rossi 2004).

11 I focus here on strategic interactions between suppliers in an opened technology system, rather than possible interactions with other technologies altogether.
case in turn.

**Incentives of Entering Suppliers**  Greater openness may in fact stimulate the individual investment incentives of outside suppliers entering into the opened technology. Additional entry may, for example, directly stimulate more investment in cases where there are strong bandwagon effects and strategic complementarities acting between entering suppliers. In such cases entry and investment may even become intertwined with diversity effects and become part of a larger virtuous circle of entry, adoption and investment (Church and Gandal 1992; Grindley 1995; Parker and Van Alstyne 2005). Absent complementary relationships, even some measure of competition and rivalry with increased entry may stimulate investment (Aghion, et al. 2005). Where opening reduces the original innovator’s control, the reduced threat of hold-up and lock-in may create further incentives for entrants to invest (Katz and Shapiro 1986; Varian and Shapiro 1998). However, even with such stimulating incentive effects and diversity effects, greater openness and entry will tend strategic interactions between entering suppliers towards competitive crowding and stifled investment incentives (Ellison and Fudenberg 2003; Augereau, Rysman et al. 2004; Economides and Skrzypacz 2004).\textsuperscript{12,13} Therefore, unless the industry structure is such that compelling complementarities between entry and investment allow intensified competition and the crowding out of incentives to be avoided,\textsuperscript{14} increasing openness and entry will eventually lead to lower investments.

Absent all other considerations (i.e. the incentives of the original innovator and coordination concerns, to follow), and to the extent that low-coordination innovations are implemented by entering suppliers rather than the original innovator the response of low-coordination innovations to openness may be summarized as follows:

**Claim 1** **Hypothesis 1** Low-coordination innovations may initially increase with greater openness, but will eventually decrease at high openness—unless there are strong strategic complementarities acting between the entry and the investment incentives.

\textsuperscript{12}Such a trend would be exacerbated by accompanying losses in bargaining power vis-à-vis the original innovator, as greater numbers of suppliers make any one supplier less critical (Brynjolfsson and Bakos 1993; Stole and Zwiebel 1996; Rajan and Zingales 1998).

\textsuperscript{13}Therefore, while “commoditizing the complementor” (Varian and Shapiro 1999) may be a means of extracting greater rents by the original innovator, it may extinguishing investment incentives of the complementor firms.

\textsuperscript{14}There may be other means in which competitive crowding can be avoided. For example, “user-innovators” may retain relatively high incentives because of their ability to directly consume their innovations.
Incentives of the Original Innovator  If a technology is partially opened, leaving the innovator with control over some critical subset of the technology, the response of the innovator to an exogenous\textsuperscript{15} increase in openness should generally not be the same as the response of entering suppliers. The innovator’s incentive response may be partially correlated those of the entering suppliers if the productivity of the original innovator’s investments increase with the investments of entering suppliers (generating greater incentives to make those investments). For example, it may be more productive to invest in an operating system if there is also high investment in complementary hardware and software components (and \textit{vice versa}). However, in retaining some measure of control over the technology, the innovator will find itself in a tug-of-war of sorts with entering suppliers to split profits. The devolution of control and promotion of entry to a system will affect bargaining power and the split of profits between entering suppliers and the original innovator (Grossman and Hart 1986; Brynjolfsson and Bakos 1993). The response of bargaining power to openness will work in opposite directions for the original innovator as for the entering suppliers, leading to a more ambivalent response to increasing openness.

### 3.2.3 Coordination and Governance

Coordination problems may be particularly trying in open technologies with multiple components and sometimes sprawling, complex systems involving economic and technical interactions across firms (Greenstein 1996). “Modularizing” the technology into separable components may reduce interactions (Baldwin and Clark 2003), but the pursuit of parochial interests by independent suppliers,\textsuperscript{16} coordination problems of the left-hand-not-knowing-what-the-right-hand-is-doing sort and other sources of coordination failure are likely to persist, nonetheless. At least \textit{a posteriori} it appears that opening a technology to multiple suppliers creates a challenge of maintaining “coherent” development (Gawer and Cusumano 2002) across multiple components and suppliers, where “unified technical leadership allow[s] within-the-firm coordination... yield[ing] a better coordination mechanism than one cutting across selling firms” (Bresnahan

\textsuperscript{15} Although openness should, in principle, be determined by the original innovator, this paper focuses on the relationship between openness and innovation \textit{per se}.

\textsuperscript{16} Research into open source software and community development suggests the possibility of emergent norms of cooperation, which may reduce actions taken in parochial interests in an open technology (Gambardella and Hall 2005).
leading to “integration and coordination benefits” (Evans, et al. 2004 p. 6) from closed organization.

Even absent direct authority and control, an innovator might still exert indirect influence across firm boundaries. Control over a critical piece of the technology, such as a computer operating system platform or essential bottleneck facilities in a telecommunications network for example, confers not only the ability to regulate access to a system but also to set the terms of access (Rochet and Tirole 2005), offering an opportunity to indirectly govern the conduct of entering suppliers. Appropriability deriving from control over critical assets also provides the innovator with the incentives to internalize externalities across the system and across time, “sponsoring” the technology system (Katz and Shapiro 1994). Critical assets that embody technical “rules” also provide some scope to regulate freedoms and restrictions and otherwise contour the costs the conduct of entering suppliers (Lessig 2000; Henderson and Kulatilaka 2003). The innovator’s devolving control and granting access to increasing numbers of suppliers may also reduce the original innovator’s ability to indirectly govern economic activity in the system.

The general tendency for greater openness to work against coordination has led researchers to argue that “systemic” innovations involving difficult coordination problems, possibly across multiple components, will be carried out in a single, integrated organization rather than across firm boundaries in an autonomous fashion (Langlois and Robertson 1989 p.364; Teece 1996). If the greater (indirect) coordinating and orchestrating ability of the original innovator leads high-coordination innovations to be disproportionately implemented by the original innovator, these innovations may also be subject to the ambivalent incentive response to greater openness of the original innovation, earlier discussed.

**Hypothesis 2** High-coordination innovations will have an ambivalent (and possibly negative) response to increasing openness.
3.3 Competition and Industry Dynamics in Handheld Computers and Smartphones

After more than a decade of technical experiments in pocketable programmable devices that could be likened to advanced calculators, a new generation of sophisticated systems emerged in the late 1980s and early 1990s, increasingly based on then-advanced 16-bit architecture. Many began to include functions of desktop PCs and design features recognizable in today's mobile computing devices. Dozens of firms entered in this period of early commercial development, creating a range of designs and product types. Measured commercial success was achieved, with several hundred thousand units sold annually in this period. Psion, a British firm, was the industry leader in this period, designing multiple award-winning products on the basis of its SIBO operating system platform.

Figure 3-1: Industry Sales, Product Releases and Market Shares

Notes: Sales volumes estimated based on IDC and Gartner archived reports and other sources. Smartphone data only include devices using public commercial radio (cellular) networks. Non-Linux or Symbian-based i-mode sales are not included. Does not include early Sharp and Casio organizers.

Notes: Market share shown at the platform or system level, rather than the individual firms level. "Other" market share includes Newton, EO, GEOS, RIM, various embedded Linux and estimates of numerous other systems.

18Features included PC-synchronization, stylus input, and graphical user interfaces.
19Based on archived documents from IDC and Gartner and trade journal articles from the early 1990s, I estimate Psion market share to have been roughly 25% of mobile computer unit sales, edging up to roughly 40% by early 1996.
By the mid-1990s, the industry became less fragmented and the leading systems of Psion, Palm and Microsoft\textsuperscript{20} possessed the bulk of market share (see following exhibit). The market began to roughly double every year in this period, reaching just under 12 million units by 2000.\textsuperscript{21} Having witnessed the example of the personal computers industry, industry executives in this period showed considerable interest in stimulating network effects around their systems by promoting the installed base of users and suppliers of complementary goods and services (McGahan, Vadasz et al. 1997; Gawer and Cusumano 2002; Nair, Chintagunta et al. 2004). However, competition more crucially depended on developing products of increasing quality and appeal (Nair, Chintagunta et al. 2004; Yoffie, Yin et al. 2004) with increasing performance and number of features (following figure).\textsuperscript{22} A stream of ever-improving products were released (Chwelos, Berndt et al. 2004),\textsuperscript{23} churning into and out of the market at roughly 2-year intervals. Multiple products were built on top of common platforms in each system. The approach of using platforms served to reduce fixed development costs (and time) and ensured technical compatibility between devices.\textsuperscript{24} The platforms themselves were also periodically upgraded.

Product innovation manifested itself in numerous dimensions, including steadily advancing processor speed and memory, reductions in size, variations in applications and form factors, and novel features and functions. Products also improved in more subtle ways, such as improved elegance in the integration of designs and nuanced trade-offs in implementing new features and advancing performance (ex: trade-offs between implementing a color screen and power consumption). Declining production costs and the availability of superior component technologies also enabled improvements. The platforms themselves were also improved—though less continuously—with a combination of roughly annual alterations and more sporadic upgrades to achieve quantum changes.

\textsuperscript{20}I focus here on the Microsoft platforms based on Windows CE, rather than the company’s earlier mobile computing projects (see Boudreau 2005).

\textsuperscript{21}Average of estimates by IDC, Gartner and other sources.

\textsuperscript{22}Nair, et al (2004) find that demand for devices is mostly explained by product features. Palm executives, quoted in the study of Yoffie, et al.(2004), assert that the appeal of a device on its own outweighed the importance of any network effect in handheld computers.

\textsuperscript{23}Chwelos, et al (2004) find the annual hedonic price index in the industry declines by roughly 20% annually, while nominal prices remain relatively stable—implying large and steady advances in the products, themselves.

\textsuperscript{24}The devices interoperated with peripheral hardware, applications software, and personal computers (with which they shared synchronized data). The devices also interoperated with each other in terms of compatible file structures and the “beaming” infra-red communications to one another.
Open Systems  The opening of systems was a major element of platform supplier strategy. The multiple products built on top of platforms were often built by multiple device suppliers.\textsuperscript{25} Of 77 mobile computer operating systems platforms between 1990 and 2004 identified in this research, most were supplied by a single supplier at the start of the 1990s, whereas roughly half were supplied by multiple suppliers from the mid-1990s, onward, including each of the market-leading systems. Moreover, each of the leading systems innovators began as the innovator of their entire systems and chose to some measure of control and grant access to outside suppliers over time.

For example, Psion began inviting outside development of devices in 1996, after over a year of integrated, closed development. Palm worked with outside value-added resellers, beginning in 1997, and with full-fledged device developers starting in 1998. Microsoft launched its first Windows CE-based platform in 1996 in partnership with a group mostly consisting of its largest PC OEM partners. Apart from different timing of opening, the degree of openness varied. The amount of control maintained over each system (in terms of giving up control over certain design and production tasks or even direct ownership in the platform) and the extent to which entry was encouraged or restricted (through a variety of technical and contractual measures)

\textsuperscript{25}Other elements of the system were also opened, such as application software.
varied over time. Competition thus essentially took the form of competing groups of firms–platform suppliers and multiple device suppliers–associated with different systems (Gomes-Casseres 1997).

Distinct Strategic Interactions in Smartphone and Handheld Computer Submarkets While modes of innovation and organization on platforms persisted past 2000, early in the new decade growth in handheld computers began to subside and a distinct submarket for smartphones emerged. Although smartphones and other devices using cellular networks had first been launched in the early 1990s, it was only by this time that a confluence of technology advances and the worldwide deployment of “third generation” cellular networks capable of efficient packet-data transmission gave this submarket considerable impetus. Sales in the smartphone submarket quickly accelerated past sales of non-networked handheld computers in just a few short years (see earlier figure). At first, device suppliers often took handheld computer platforms and built smartphones on them. Later, the platforms themselves were altered by the platform suppliers, themselves, to facilitate smartphone development. Most starkly, Psion placed its platform assets into a new joint venture, Symbian, in which Psion’s share was diluted to 40%, with entering device suppliers capitalizing the remaining costs to convert Psion’s platform to a specialized smartphone platform.

Notwithstanding the overlap in technology and modes of organization in handhelds and smartphones, market dynamics in the two submarkets were rather different. In handhelds, modest network effects may have generated some measure of complementarities between suppliers, where entry of device suppliers stimulated with market for all devices in a system. But the products of different suppliers were, for the most part, substitutes (and certainly so, on the margin). For example, Psion and Palm shared an abiding concern for lost market share of their device-building businesses which kept them from opening more widely or earlier than they did.26 Also, device suppliers such as Sony (in the Palm system) and Toshiba (in the Microsoft system) made quite explicitly reduced their investments and even exited in times when competitive crowding increased in their systems.

In contrast, strategic interactions among suppliers of smartphones in the same systems

26Interviews with both ex-Psion and Palm executives (2004). Butters and Pogue (2002) also detail Palm executives’ concerns that wide licensing would cannibalize the revenues of Palm’s device-building business.
have been much more complementary—at least in the sample period observed to date. This has related to widespread belief that the smartphone submarket is subject to increasing returns and winner-take-all dynamics, where gaining acceptance by a broad set of suppliers, and device-builders in particular, may be critical to the success of a system (Eisenmann and Suarez 2004). Accordingly, considerable effort has been exerted by platform suppliers to promote entry of device suppliers to their systems, with investments in technical support, business development resources and outright attempts to woo marquee device suppliers.

These contrasting market dynamics—high complementarities in smartphones and low complementarities in handheld computers—can be readily observed in regressions of product development decisions of individual firms in response to increasing entry. High levels of entry lead suppliers to reduce product development in handheld computers. In contrast, high levels of entry lead individual suppliers to increase product development in smartphones (see Appendix Table 6).

3.4 Data Set

An unusually rich data set was compiled. The data used in this study come from three main sources. A comprehensive data set of products in each system was used to construct measures of innovation. Measures of openness across systems and over time were constructed on the basis of detailed review of the industry and technologies on the basis of interviews and secondary sources. Market data was mainly compiled from archival materials of IDC and Gartner Group. Additional variables were derived from these data sets.

3.4.1 Measuring Innovation

A comprehensive list of product releases for each device supplier in each system was collected, including product name, device supplier name, platform/system name, release date. Each prod-

---

27 Whether or not the market will in fact “tip” to a given system or another remains to be seen. But the need for network operators and complementary suppliers to make specific sunk costs in a system, and the growing importance of interoperability in a networked context, the potentially staggering size of the market (current mobile phone shipments are nearing 1 billion units per year) and accompanying strategic uncertainty relating to the decisions of suppliers are each consistent with this thesis.

28 Interviews with multiple industry executives and analysts.
uct was coded in relation to 58 measures of performance and characteristics (see appendix). Data were compiled by hand from current and archived websites of platform owners and device suppliers, product listings from point of sale data from NPD Intellect, current and archived catalogues, trade publications and user magazines, web sites of interest groups and hobbyists, articles from business and industry press on Lexis-Nexus, and by communicating with the companies, themselves. Continuous measures of performance were used to construct measures of the advancing frontier in each system as a whole, by tracing the leading performance levels of all products in a system within the dimension of interest up to a point in time. Discrete measures of new features, coded by an indicator variable (1-0), lent themselves to the construction of counts of new features. Derivative measures, such as “new combinations” of already-existing features and performance levels were also constructed.

Most discrete measures (i.e. 1-0) of new features represented innovations “on top” of the platform most often implemented by device suppliers. These included features of industrial design, applications, peripheral hardware and the like. Exceptions to this were changes in low-level system software and architecture, such as those associated with new communications capabilities or ports and connectors. Therefore, most discrete measures were treated as low-coordination innovations, with exceptions treated as high-coordination innovations.

Most continuous measures of performance—such as processor speed, RAM memory, pixel count, color depth—were implemented through changes in the low-level design of the system, often residing in system software or basic architecture. In the majority of instances, these were implemented through changes in the platform, in which case corresponding changes in design often needed to be made by device designers to realize these advances. Continuous measures that did not cleanly fall into the high-coordination category were changes in physical volume, and the ratio of pixels to the physical area of the device (and indication of the “tightness” of design). Therefore, most continuous measures were treated as high-coordination innovations.

29 The measures chosen included those that have been used in other studies of the mobile computer industry (Chwelos, Berndt et al. 2004; Nair, Chintagunta et al. 2004) and other measures.
30 The isolation of individual dimensions of performance as measures of innovation comes with obvious drawbacks, such as: 1) all advances may not be equally valuable; 2) performance across dimensions may be related by inherent design tradeoffs; 3) innovation often takes the form of increasing elegance and “tightness” of integration; 4) measures of performance may not necessarily be directly comparable across systems (ex: processor speed). It appears these limitations have been adequately attended to in the approach that follows to still yield meaningful patterns.
3.4.2 Measuring Openness as Granting Access and Devolving Control

In partially open technical systems, the extent to which control is devolved may vary independently of the extent to which access is granted to outside suppliers (Section 2). Therefore, separate measures are developed to reflect each of these dimensions of openness.

Multiple candidate measures of control maintained by the original innovator (who becomes a platform supplier) were collected. These included 1) the number of components and sub-tasks falling under the exclusive control of the platform supplier (based on a list of 23 components intended to be comprehensive of a system), rather than opened to device suppliers; 2) the number of such components and sub-tasks supplied by the original innovator, but not exclusively so; and 3) concentration of share ownership in the original innovator, cum platform supplier. These measures were coded on a monthly basis for the sample years on the basis of company press releases, trade magazine articles, archived company websites and interviews with company executives and industry watchers. Control measures were multiplied by negative one so increasing values could be interpreted as greater openness or devolving control.

The measure producing most statistically significant results is the number of components under the control of the platform supplier; the number supplied does not produce significant results. Share ownership produced weak results. Interactions of share ownership with scope of control appear to gain slightly more statistical significance over scope of control on its own, perhaps by producing slightly more variation while both measuring control.

The granting of access to the platform to device manufacturers (and otherwise encouraging or discouraging entry) was measured by constructing a score of contracting and technology decisions taken by platform suppliers. The strategies of platform suppliers were coded 1-0 for multiple issues related to the liberalness (restrictiveness) of licensing, the provision of technical support to enable entry, the scope for differentiation enabled by the platform, and other efforts to promote entry by device suppliers. The full range of criteria allowed a maximum score of 15 (“high access”). No attempt was made to weight this score. Again, the score was developed on the basis of press releases, industry press articles, archived company websites and interviews.
3.4.3 Other Data and Variables

Data were also collected on industry structure and other controls to account for alternative determinants of innovation outcomes. Principal among these were measures of market size in terms of industry and system unit volumes for both handheld computers and smartphones. The primary estimates of unit sales from 1990 to 1995 were provided by IDC from their archives of reports from that period; later industry sales estimates to 2004 were based on numbers from Gartner's quarterly report of unit sales for the industry.

3.5 Empirical Strategy and Model

The relationship between openness and innovation is investigated in a basic reduced-form framework, as follows, where the relationship between innovation, $y$, and openness, $O$—given by $f$—is assumed to be linearly separable from alternative determinants of innovation, $X$, and a stochastic error term, $\varepsilon$. Unobserved variables impacting innovation are collected under $\Psi$. Arguments are indexed by the technology system, $i$, the dimension of innovation $m$, and the time period, $t$.

The use of multiple measures of innovation implies essentially the same model is run multiple times with different dependent variables to see if patterns between innovation and openness are found and these these patterns are regular across multiple innovation measures.\(^{31}\) This has the added advantage of providing a sense of the robustness of results.

\[
y_{imt} = f(O_{it}; \Theta) + \beta_X \cdot X_{imt} + \Psi_{imt} + \varepsilon_{imt} \tag{3.1}
\]

Following the earlier hypotheses, the heart of the empirical strategy is to examine how the relationship, $f$, varies with structural conditions, $\Theta$. The structural conditions hypothesized to crucially affect the relationship were the nature of innovation (high coordination versus low coordination tasks) and the nature of strategic complementarities acting between entering devices suppliers (high complementarities versus low complementarities, as found in smartphone and handheld computer submarkets, respectively). This essentially produces four cases, $\Theta \in$

---

\(^{31}\) Although modeling of concurrent dimensions of innovation suggests the possibility of gaining efficiency by regressing the multiple equations as a simultaneous system, the use of same explanatory variables in each models means there will be no gain of efficiency in doing so (Greene 2000).
TABLE 1: Variable Definitions and Descriptive Statistics

<table>
<thead>
<tr>
<th>Variable Description</th>
<th>N</th>
<th>Mean</th>
<th>StdDev</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variables: Low Coordination Innovations (&quot;Modular,&quot; &quot;Autonomous&quot; Innovations)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New product release count</td>
<td>156</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>Novel products</td>
<td>156</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>New combination count</td>
<td>156</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>New feature count</td>
<td>156</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Physical volume</td>
<td>156</td>
<td>10</td>
<td>5</td>
<td>1</td>
<td>27</td>
</tr>
<tr>
<td>Processor speed</td>
<td>156</td>
<td>168</td>
<td>162</td>
<td>8</td>
<td>624</td>
</tr>
<tr>
<td>RAM memory</td>
<td>156</td>
<td>39</td>
<td>41</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Color depth</td>
<td>156</td>
<td>11</td>
<td>7</td>
<td>1</td>
<td>18</td>
</tr>
<tr>
<td>Pixel count</td>
<td>156</td>
<td>252</td>
<td>238</td>
<td>26</td>
<td>786</td>
</tr>
<tr>
<td>Pixels per area</td>
<td>156</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>33</td>
</tr>
<tr>
<td><strong>Measures of Openness (&quot;Control&quot; and &quot;Access&quot;)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scope of control</td>
<td>156</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>Scope of supply</td>
<td>156</td>
<td>15</td>
<td>3</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>Concentration of ownership</td>
<td>156</td>
<td>91</td>
<td>22</td>
<td>25</td>
<td>100</td>
</tr>
<tr>
<td>Control</td>
<td>156</td>
<td>1185</td>
<td>568</td>
<td>152</td>
<td>1900</td>
</tr>
<tr>
<td>Access</td>
<td>156</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td><strong>Measures of Entry and &quot;Within-System&quot; Market Structure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of suppliers</td>
<td>156</td>
<td>9.9</td>
<td>11.5</td>
<td>1</td>
<td>46</td>
</tr>
<tr>
<td>Suppliers/units</td>
<td>156</td>
<td>0.02</td>
<td>0.04</td>
<td>0.0006</td>
<td>0.2</td>
</tr>
<tr>
<td>Suppliers/year change</td>
<td>156</td>
<td>2</td>
<td>4</td>
<td>-8</td>
<td>16</td>
</tr>
<tr>
<td><strong>Other Control Variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smartphone</td>
<td>156</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Unit shipments</td>
<td>156</td>
<td>2059</td>
<td>2872</td>
<td>0</td>
<td>14400</td>
</tr>
<tr>
<td>Unit shipments, total system</td>
<td>156</td>
<td>3365</td>
<td>3243</td>
<td>0</td>
<td>14400</td>
</tr>
<tr>
<td>Unit shipments, total submarket</td>
<td>156</td>
<td>6517</td>
<td>5655</td>
<td>45</td>
<td>17400</td>
</tr>
<tr>
<td>Unit shipments, total industry</td>
<td>156</td>
<td>12234</td>
<td>9026</td>
<td>684</td>
<td>29687</td>
</tr>
<tr>
<td>Time</td>
<td>156</td>
<td>66</td>
<td>8</td>
<td>37</td>
<td>84</td>
</tr>
</tbody>
</table>

Other control variables include platform fixed effects, lagged dependent variables, year effects, lagged supplier counts.
**TABLE 2: Two-Way Correlation Matrix of Main Dependent and Explanatory Variables**

### Unaltered Variables

<table>
<thead>
<tr>
<th></th>
<th>New product release count</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novel products</td>
<td>.9*</td>
<td>1.00</td>
</tr>
<tr>
<td>New combination count</td>
<td>.9*</td>
<td>1.00</td>
</tr>
<tr>
<td>New feature count</td>
<td>.2*</td>
<td>.14</td>
</tr>
<tr>
<td>Physical volume</td>
<td>.2*</td>
<td>.28*</td>
</tr>
<tr>
<td>Processor speed</td>
<td>.13</td>
<td>.06</td>
</tr>
<tr>
<td>RAM memory</td>
<td>.08</td>
<td>.18*</td>
</tr>
<tr>
<td>Color depth</td>
<td>-.05</td>
<td>.01</td>
</tr>
<tr>
<td>Pixel count</td>
<td>.3*</td>
<td>.31*</td>
</tr>
<tr>
<td>Pixels per area</td>
<td>.07</td>
<td>.15</td>
</tr>
<tr>
<td>Scope of control</td>
<td>-.3*</td>
<td>-.21*</td>
</tr>
<tr>
<td>Scope of supply</td>
<td>.39*</td>
<td>.25*</td>
</tr>
<tr>
<td>Concentration of ownership</td>
<td>-.17*</td>
<td>-.12</td>
</tr>
<tr>
<td>Control</td>
<td>-.36*</td>
<td>-.23*</td>
</tr>
<tr>
<td>Access</td>
<td>.22*</td>
<td>.28*</td>
</tr>
<tr>
<td>Unit shipments</td>
<td>.34*</td>
<td>.24*</td>
</tr>
<tr>
<td>Unit shipments, total</td>
<td>.28*</td>
<td>.41*</td>
</tr>
<tr>
<td>Time</td>
<td>.23*</td>
<td>.21*</td>
</tr>
</tbody>
</table>

### Variables with Platform Fixed Effects and First and Second Order Time Trends Removed

<table>
<thead>
<tr>
<th></th>
<th>New product release count</th>
<th>1.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novel products</td>
<td>.35*</td>
<td>1.00</td>
</tr>
<tr>
<td>New combination count</td>
<td>.71*</td>
<td>1.00</td>
</tr>
<tr>
<td>New feature count</td>
<td>.39*</td>
<td>1.00</td>
</tr>
<tr>
<td>Physical volume</td>
<td>.23*</td>
<td>.22*</td>
</tr>
<tr>
<td>Processor speed</td>
<td>.09</td>
<td>.09</td>
</tr>
<tr>
<td>RAM memory</td>
<td>.00</td>
<td>.19</td>
</tr>
<tr>
<td>Color depth</td>
<td>-.02</td>
<td>-.05</td>
</tr>
<tr>
<td>Pixel count</td>
<td>.34*</td>
<td>.28*</td>
</tr>
<tr>
<td>Pixels per area</td>
<td>-.02</td>
<td>.09</td>
</tr>
<tr>
<td>Scope of control</td>
<td>-.07</td>
<td>.00</td>
</tr>
<tr>
<td>Scope of supply</td>
<td>.18*</td>
<td>-.02</td>
</tr>
<tr>
<td>Concentration of ownership</td>
<td>-.08</td>
<td>-.07</td>
</tr>
<tr>
<td>Control</td>
<td>.00</td>
<td>.13</td>
</tr>
<tr>
<td>Access</td>
<td>.07</td>
<td>.04</td>
</tr>
<tr>
<td>Unit shipments</td>
<td>.03</td>
<td>.06</td>
</tr>
<tr>
<td>Unit shipments, total</td>
<td>.13</td>
<td>.29*</td>
</tr>
</tbody>
</table>

* significant at 5%

Figure 3-4: Two Way Correlations Between Main Variables
The following figure restates the Hypotheses 1 and 2 explicitly showing the empirical strategy. The response of low-coordination innovations will vary according to the nature of strategic interactions acting between suppliers, given by the industrial context. Where crowding-out of incentives can be avoided because of strong complementarities between entry and individual investment incentives, the relationship is predicted to be positive—otherwise the relationship will turn negative at least at high openness and entry. High-coordination innovations are predicted to respond ambivalently to increasing openness. To capture these possible patterns, the relationship is specified to allow the relationship to vary with the context, \( f(O; \Theta) = \beta_A(\Theta) \cdot \text{ACCESS}_{it} + \beta_C(\Theta) \cdot \text{CONTROL}_{it}, \) for each of the measures of openness. Quadratic transforms of openness measures are also included in certain regressions to capture curvilinear relationships.

Figure 3-5: Restatement of Hypotheses in Terms of the Empirical Strategy

The particular model specification in the regressions will deviate from the linear set-up according to the nature of the dependent variable being modeled. Most dependent variables of interest are either non-negative count variables (in cases of countable innovations) or non-negative continuous variables (in cases of advancing performance frontiers). In the case of
count variables a count framework was used with appropriate distributional assumptions and accounting for heteroskedasticity with numerous specifications, although only negative binomial regressions are reported herein. In the case of non-negative continuous variables, a Tobit specification, censored at zero is employed. Exceptions are noted in Section 6.

3.5.1 Identification Issues

Potential endogeneity problems are the main challenge to identifying the openness-innovation relationship. Certain such problem, such as simultaneous trending may be dealt with by including time trends. Other such problems such as the possibility of mutual causation, where openness not only affects innovation but the state of technology also systematically affects openness can be cautiously ruled-out.

The main challenge to measuring the primary coefficients of interest, $\beta_A$ and $\beta_B$, without bias is the possibility that omitted variables, part of $\Psi$, are correlated with both the openness variables and the dependent variable, resulting in a spurious correlation. While the extraordinary pace and uncertainty of industry change and slow and blunt implementation of openness policies appear to have generated a great deal of exploitable variation in these variables, unbiased identification of coefficients requires the relationship be identified with only exogenous variation in openness variables. A failure to control for unobserved “good news” (bad news) affecting a technical system might, for example, reduce (increase) the innovator’s willingness to give up appropriability and open the system while increasing (decreasing) innovation, resulting in a downward bias in coefficient estimates.

A first step to isolating exogenous variation is to include a number of control variables for alternative determinants of innovation and openness, including those reflecting market opportunity, technical opportunity, strategic interactions with other systems and other dimensions.

---

32 The cross section of systems is too narrow to estimate time dummies without bias and so a quadratic time trend is employed instead.

33 Regressions of the openness variables on lags of numerous measures of technology levels finds no correlation. Further, the sample period was explicitly chosen to avoid early time periods when limits in the technology factored into decisions to open.

34 Secondary material and interviews with industry executives suggest substantial second-guessing and backtracking on openness strategies. This is not surprising given the limited history of open strategies used in the economy.
of industry structure. The analysis then also exploits the panel structure of the data to minimize omitted variable problems. Non-transitory (cross-sectional) components of unobserved heterogeneity are addressed with individual fixed effects for the system and dimension of innovation, $\eta_{im}$. Remaining unobservable variation that may lead to spurious correlation will pertain to unobserved transitory variation that is specific to a given technical system. Lags of the dependent variable and counts of the number of suppliers in a system, $N$, are included to "soak up" as much of this variation as possible. The model used is therefore as in the following expression, where the error term has been redefined as $\epsilon_{int} = \psi_{int} + \eta_{int}$, where $\psi_{int} = \psi_{int} - (\delta_{int} + \alpha y_{int(-t)} + \gamma N_{i(t-1)} + \eta_{int})$. Meaningful coefficients rely on the correlation between openness variables and $\psi$ to be relatively small.

$$y_{int} = \beta_A (\Theta) \cdot ACCESS_{it} + \beta_C (\Theta) \cdot CONTROL_{it} + X_{int}' \beta_X +$$

$$+ \delta_{int} + \alpha y_{int(t-1)} + \gamma N_{i(t-1)} + \eta_{int} + \epsilon_{int}$$

Apart from these endogeneity challenges, there is also a selection problem of sorts, as the coefficients will reflect the part of the curve that is observed in the data. In the case of low-coordination and low-complementarities, for example, I predict a non-monotonic relationship between openness and innovation. In such a case, the observed slope may be positive, negative or even zero, depending on how openness is set. If it is costly to implement greater degrees of openness or other interests in setting the degree of openness (such as promoting bandwagon effects), the level of observed openness may be systematically biased in one direction or another. Special care will therefore be taken in interpreting coefficients.

Finally, having severely distilled firm and product-level data to the level systems, the data

---

35 Vector $X$ includes multiple measures of market size, growth, market share, cumulative past product releases, platform age and upgrade versions.

36 Several instrumental variables were considered, including variation of technological advance in competitor systems (for cases where the innovation is not a general innovation or one subject to spillovers), increasing knowledge about open strategies (measured by citations of articles and books on the topic).

37 The introduction of the lagged dependent variable creates another possible source of endogeneity. If there is autocorrelation in errors the lagged variable will be directly correlated with the error term. Without autocorrelation, the lagged dependent variable is still not be strictly exogenous in that it is correlated with future values and therefore is correlated with the average error in cases where there are fixed effects (Arellano and Bond 1991), a problem that may be smaller in a long panel as the one used here.
will be pooled in several ways and coefficients identified with interaction terms in hopes of better exploiting the data and increasing the efficiency of estimates.

3.6 Analysis and Results

3.6.1 The Response of Low-Coordination Innovations to Openness

Table 3 reports results of regressions of multiple low-coordination innovations on openness measures, controlling for other factors, for both handheld and smartphone submarkets. The regularities found across the multiple regressions conform to predictions. The multiple dependent variables appearing in the results are: new product releases, novel products, new combinations, new features count, decreases in device size, pixels per physical size. In the case of count dependent variables, results for negative binomial models with system fixed effects are reported (see Cameron and Trivedi 1998). In the case of modeling advances in continuous variables, results for Tobit model specifications (see Wooldridge 2000) are reported.

Given limited degrees of freedom in the smartphone submarket, the approach used to identify coefficients is to pool the data for handhelds and smartphones, and then identify the different responses to openness by including separate terms that interact openness variables with dummy variables of the smartphone submarket. Apart from possible efficiency gains, this approach provides a direct means of testing whether the coefficients on the openness variables in different submarkets are different from one another.

A regular pattern immediately emerges across regressions of these alternative measures of innovation on measures of openness for the different submarkets (Models 3-1 to 3-6). The coefficient on GRANTING ACCESS (corresponding to the handhelds submarket) is not statistically different from zero. However, granting access interacted with a dummy for the smartphone submarket, ACCESS x SMARTPHONE, is positive and significant at the 1% level in five of the six regressions of measures of low-coordination innovations, and significant at the 15% level in the weakest case (decreases in volume, the dependent variable with the least variation). The

---

38 Given the use of the same set of regressors across models for each measure of innovation, assumptions of correlation in the error terms cannot be used to improve the efficiency of coefficient estimates.

39 Numerous specifications were assessed, including Poisson, linear, and attempts to correct for markedly underdispersed data with an error modeled with a first stage Pearson residual. Each produced roughly the same results.
coefficient on DEVOLVING CONTROL (corresponding to the handhelds submarket) is positive and significant in three of six cases (significant at the 1% level for novel products and new combinations, 5% for new product release counts). The interaction of control with smartphone, CONTROL x SMARTPHONE is positive and significant in the same cases.

Broadly, these results suggest that devolving control is positively related with low-coordination innovations in the case of handhelds, and more positively in the case of smartphones. The first order relationship between low-coordination innovations and access, however, was found to be positive only in the case of the smartphone market; in the case of handhelds, the relationship appears to be zero. The uniformly positive set of relations between low-coordination innovations and openness in the case of the smartphone data are consistent with predictions. The results for handheld computers require additional analysis and interpretation to determine whether they are consistent or not with predictions.

To determine whether the relationship between granting access and innovation in handheld computers is truly zero, whether there is simply not enough variation to identify the coefficient, or perhaps the first order model simply fails to capture a more complex relationship, the innovation data from the first six regressions was pooled in a single regression (Model 3-7). The intent in doing so was to create more useful variation to identify the nature of the relationship between openness and innovation. To attempt to then minimize the additional noise brought by combining different types of innovations, individual dummies and time trends for each measure of innovation were included in the regression. To simultaneously model count and continuous measures of innovation, a log-linear specification was used where positive values were converted to their logarithm, and zero values were maintained as zeros. Standard errors were clustered by the innovation type and data were inversely weighted with the size of their residual in a first-stage regression.

The results from this approach produce the very same set of coefficient signs on first order and interacted access and control measures but with higher statistical significance (although the difference between coefficients on measures of control across submarkets was no longer statistically significant), suggesting the strategy has worked (Model 3-7). The results offer further evidence the first order relationship between low-coordination innovation and GRANTING ACCESS is zero.
To more directly study the relationship between access and low coordination in innovation, the data for smartphones was dropped, but data of multiple innovation measures kept pooled. The second order relationship between GRANTING ACCESS and pooled low-coordination innovations was found to be highly significant at the 1% level for each of the individual first and second order terms. The coefficient on DEVOLVING CONTROL persisted as positive and highly significant and was estimated as almost precisely the same magnitude as in the earlier pooled regression (0.0024 versus 0.0023). It appears that the average level of GRANTING ACCESS (5.0) is near the estimated location of the peak of the relationship between low-coordination innovations and GRANTING ACCESS (4.1), based on pooled data regressed on just the handheld computer data (Model 3-8). We would expect that prior to rising prominence of smartphones (wherein opening wide to promote bandwagon effects and adoption was increasingly important) that openness would be set at the peak or at levels lower than the peak to the extent that implementing openness is costly. This is indeed what is observed, as GRANTING ACCESS is at an average level of 3.2 prior to 2002, after which time strategies more explicitly geared to the smartphone submarket.

The relationship between low-coordination innovations and DEVOLVING CONTROL is positive, but an inverted-U relationship cannot be observed explicitly. This is consistent with predictions if control is being given up in relatively low amounts compared to how much access is being granted and is therefore set “to the left” of a peak that is not observed in the data. If giving up control is less costly (either to implement or in terms of lost appropriability of the original innovator), it should not be surprising that the treatment of control is more conservative. As further corroboration that the relationship between low-coordination innovations and giving-up control take the predicted shape of an inverted-U beyond the sample data, the model was re-run allowing a quadratic relationship with DEVOLVING CONTROL. Consistent with predictions, the relationship was found to be concave—although the second order coefficient was not statistically significant (not reported).

Each of these patterns was generally robust to including and excluding lagged variables and dynamic panel specifications (Arellano and Bond 1991), variations in the control variables used, excluding either early or late years in the sample, replacing time trends with time fixed
The coefficients imply a strong relationship between openness and innovation. Focusing on the regression that pools the low-coordination innovations (Model 3-7), the coefficients or even their implied incidence rate ratios are difficult to directly interpret. (For example, the coefficient on DEVOLVING CONTROL is 0.002, implying an incidence ratio of just 1.002). Therefore, I consider the magnitude of impact of one (within-system) standard deviation change of the openness measure. Using this logic, one standard deviation change in DEVOLVING CONTROL corresponds with a 29% change in low-coordination innovations in handheld computers and a 48% increase in low-coordination innovations in smartphones. The apparent sizeable effect of devolving control is commensurate with the prominence of openness strategies in the industry; the greater impact of openness in smartphones is consistent with greater complementarities acting in that submarket. Further, it should be cautioned that these represent first order and local measurements of the slope; if the true slope is concave (as is predicted in the case of handheld computers), these values may over-estimate the effect at higher levels of openness.

Following the same approach, an increase of one standard deviation in GRANTING ACCESS corresponds with an increase of low-coordination innovations of 264%. This seems quite high, but GRANTING ACCESS (by the definition of the variable) measures direct attempts to affect entry and the number of suppliers in a system; the large effect appears commensurate with the strong, seeming multiplicative effect on the generation of product variations and new features that came about with the multiplication of suppliers in a system.

This approach to assessing the magnitude of impact cannot be used in the case of granting access in the handheld submarket, as the first order effect was zero. Instead the slope of the relationship is assessed at an arbitrarily low point (one sample standard deviation less than the sample mean) and an arbitrarily high point (one sample standard deviation greater than the sample mean). The local slope at these points is interpreted as the percentage change that would result from a (within-system) standard deviation change in GRANTING ACCESS at those points. The changing magnitudes in response—moving from +30% impact at the low level

---

10 Fixed time effects are a more stringent control but may be biased when estimated on a thin cross-section, as used here.
11 The mechanisms through which granting access and devolving control act are analyzed in the following section.
to roughly -50% impact at the high level—suggests a relatively sharp inverted-U relationship.

3.6.2 The Response of High-Coordination Innovations to Openness

Table 4 reports results of regressions of multiple high-coordination innovations on openness measures, controlling for other factors, for both handheld and smartphone submarkets. Results are consistent with predictions; although there is some evidence that the frequency of innovations increases, the rate of advance of performance is not statistically related to openness. The multiple dependent variables used are: processor speeds, RAM memory, pixel counts, increasing color depth and counts of new communications and system functions involving low-level system functionality. Just as in the earlier discussion, results negative binomial and Tobit specifications are reported. The regressions also pool data for handhelds and smartphones and identify distinct effects for handhelds and smartphones using a dummy variable for the smartphone submarket interacted with each openness variable.

In contrast to the striking regularities in low-coordination regressions (repeated in the first column of Table 4), regressions of changes in the advancing frontier of high-coordination innovation measures were not statistically different from zero in most cases (Models 4-1 to 4-4). Where there were patterns, these were far less striking and often diverged from patterns in low-coordination innovations. For example, the relationship between high-coordination innovations and GRANTING ACCESS (not interacted, therefore relating to the handheld submarket) is negative in each case, and statistically significant in the case of processor speed advances (Model 4-1). The relationship with GRANTING ACCESS is more positive in the case of the smartphone submarket (positive across Models 4-1 to 4-4), consistent with greater complementarities in that market. But for the most part these are not statistically significant, consistent with a more ambivalent response of high-coordination innovations to greater openness.

One possible reason for the lack of observed correlation is there may simply be inadequate degrees of freedom and variation in the data to detect trends. To offer further corroborating evidence, more discerning tests or corroborating evidence would be useful. Unfortunately, the pooling of multiple measures of innovation to create greater degrees of freedom, as was used in low-coordination innovations, cannot as readily be applied in this set of innovation measures that diverge to greatly in scale and patterns. To determine whether control variables
TABLE 3: Low-Coordination Innovations Regressed on Openness, System-Level Data

<table>
<thead>
<tr>
<th>Sample: Handheld and Smartphone Data, Pooled (N=156)</th>
<th>Handheld Data (N=780)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dependent Variable:</strong></td>
<td></td>
</tr>
<tr>
<td>New Products</td>
<td></td>
</tr>
<tr>
<td>Novel Products</td>
<td></td>
</tr>
<tr>
<td>New Combin's</td>
<td></td>
</tr>
<tr>
<td>New Feature Count</td>
<td></td>
</tr>
<tr>
<td>Decreases in Device Size</td>
<td></td>
</tr>
<tr>
<td>Pixels per Physical Size</td>
<td></td>
</tr>
<tr>
<td>Pooled Innovs(3)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Model:</strong> (3-1)</td>
<td></td>
</tr>
<tr>
<td>Openness Variables, Interacted with Submarket</td>
<td></td>
</tr>
<tr>
<td>Granting access</td>
<td></td>
</tr>
<tr>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>Access x smartphone</td>
<td></td>
</tr>
<tr>
<td>Devolving control(1)</td>
<td></td>
</tr>
<tr>
<td>Control x smartphone</td>
<td></td>
</tr>
<tr>
<td>Control Variables</td>
<td></td>
</tr>
<tr>
<td>Dependent variable, lagged</td>
<td></td>
</tr>
<tr>
<td>Dependent variable, cum., lagged</td>
<td></td>
</tr>
<tr>
<td>No. suppliers, lagged</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Time(2)</td>
<td></td>
</tr>
<tr>
<td>Smartphone dummy</td>
<td></td>
</tr>
<tr>
<td>Market Var(2)</td>
<td></td>
</tr>
<tr>
<td>Platform FE, dummies</td>
<td></td>
</tr>
<tr>
<td>Log-Likelihood Specification</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>FE/NB (1)</th>
<th>FE/NB (1)</th>
<th>FE/NB (1)</th>
<th>FE/NB (1)</th>
<th>Tobit(3)</th>
<th>Tobit(3)</th>
<th>FE/log-lin</th>
<th>FE/log-lin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard errors in parentheses</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>* significant at 5%; ** significant at 1%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(1) Devolving control is measured by count of number of components within the broader systems exclusively controled by the platform supplier, multiplied by concentration of ownership of the platform. This is multiplied by negative one to give a measure that is valued negatively. (2) Includes (lagged) units sold in the entire system, the submarket within a given system, the total submarket and the total industry, along with quadratic and growth terms. (3) Logarithms were taken of the individual innovation measures to combine the count measures with continuous measures. Data points were weighted by the inverse of residual from a first-stage regression. Standard errors of coefficients have been clustered. (4) Dummies and time trends for individual innovation types were included.

Figure 3-6: Low-Coordination Innovations Regressed on Openness, System-Level Data
are simply annihilating what useful variation there is, the model was re-run with different combinations and subsets of control variables. Results remained indistinguishable from zero, except where the number of controls were very dramatically reduced, in which cases regressions produced divergence coefficient signs—both across individual measures of innovation and in relation to the patterns found in low-coordination innovations (not reported). Regressions were also re-run with measures of the frequency of incidence of high-coordination innovations, rather than their magnitudes (Models 4-5 to 4-9), in which case the relationship between innovation frequencies and GRANTING ACCESS in smartphones was consistently and significantly positive—suggesting that openness may increase the frequency of advances, but the net effect on actual magnitudes is not distinguishable from zero—consistent with predictions.

3.7 Key Underlying Mechanisms

The patterns found in preceding regressions are consistent with the tensions between diversity, incentives and coordination effects with greater openness that were conjectured in Section 2. This sections provides evidence of some of the most basic assumptions underlying these hypotheses.

■ Openness Increases Within-System Entry  By the way that granting access was defined, as instruments set by the platform supplier intended to affect freedom of entry to a system, we should expect higher levels of the variable GRANTING ACCESS to systematically correspond with higher entry, controlling for other explanations for entry. Further, it was conjectured that devolving control should also lead to increased entry, all else being equal, as devolving control may reduce the threat of lock-in and hold-up to the platform supplier and possibly also increase the scope for differentiation of entrants. This is perhaps the most basic assumption and starting point for all other claims and hypotheses in this paper. To investigate whether this assumption is correct, measures of within-system entry were regressed on the openness measures, controlling for alternative determinants of entry with market controls, time trends, and system fixed effects. The model is a simple OLS specification.
### TABLE 4: High Coordination Innovations Regressed on Openness, System-Level Data

<table>
<thead>
<tr>
<th>Dependent Variable, Interacted with Submarket</th>
<th>Measures of High-Coordination Innovation</th>
<th>Frequency of Advances</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pooled Magnitude of Advances</td>
<td>Frequency of Advances</td>
</tr>
<tr>
<td></td>
<td>Handheld and Smartphone Data, Pooled (N=156)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integ'n of higher proc speed</td>
<td>Integ'n of higher RAM memory</td>
</tr>
<tr>
<td></td>
<td>(4-1)</td>
<td>(4-2)</td>
</tr>
<tr>
<td>Openness Variables, Interacted with Submarket</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Granting access</td>
<td>.04</td>
<td>-5.4*</td>
</tr>
<tr>
<td>Access x smartphone</td>
<td>.02</td>
<td>.27.2</td>
</tr>
<tr>
<td>Devolving control(1)</td>
<td>.002*</td>
<td>.000</td>
</tr>
<tr>
<td>Control x smartphone</td>
<td>.001</td>
<td>-.01</td>
</tr>
<tr>
<td>Control Variables</td>
<td>.002</td>
<td>.00</td>
</tr>
<tr>
<td>Dependent variable, lagged</td>
<td>.002</td>
<td>.00</td>
</tr>
<tr>
<td>Dependent variable, cum., lagged</td>
<td>.002</td>
<td>.00</td>
</tr>
<tr>
<td>No. suppliers, lagged</td>
<td>.002</td>
<td>.42</td>
</tr>
<tr>
<td>Time</td>
<td>.002</td>
<td>.00</td>
</tr>
<tr>
<td>Time2</td>
<td>.002</td>
<td>.20</td>
</tr>
<tr>
<td>Smartphone dummy</td>
<td>.002</td>
<td>-11.07</td>
</tr>
<tr>
<td>Log-Likelihood</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Standard errors in parentheses**  
* significant at 5%; ** significant at 1%

1. (1) Devolving control is measured by count of number of components within the broader systems exclusively controled by the platform supplier, multiplied by concentration of ownership of the platform. This is multiplied by negative one to give a measure that.

2. (2) Includes (lagged) units sold at the level of the system, the submarket within a given system, the total submarket and the total industry, along with quadratic and growth terms.

3. (3) Logarithms were taken of processor speed, RAM and pixel count given these measures grow exponentially. Color depth is already measured in terms of an exponent. Changing the specification does not change results.

4. (4) For all Tobit regressions presented here (with one exception), the dependent variable is logarithm of changes in the leading values (frontier).

Zero values are left as zero. The exception is color depth, where this measure is already given as an exponent.

5. (5) Dummies and time trends for individual innovation types were included. High coordination innovations were not similarly pooled, because of the far greater differences in magnitude and variation in that group.

---

Figure 3-7: High Coordination Innovations Regressed on Openness, System-Level Data
\[ \Delta \text{ENTRY}_{it} = \beta_A (\Theta) \cdot \text{ACCESS}_{it} + \beta_C (\Theta) \cdot \text{CONTROL}_{it} + \beta_X X_{imt} + \delta_m t + \eta_{im} + \epsilon_{imt} \quad (3.3) \]

To reduce the extent of endogeneity problems caused by simultaneous trending of openness and entry, the entry measure—the count of the number of suppliers—is differenced (annual change of firms). In the case of the relatively mature handheld market where crowding (the ratio of number of suppliers to market opportunity) is expected to be meaningful, this measure is also modeled as it may be less susceptible to trending as the absolute number of suppliers. Results are reported in Table 5. The model is run separately for the handheld computer and smartphone submarket data. In all regressions of entry variables on openness variables the coefficients were positive. These were positive and large in magnitude (on the order of 1 additional entrant or more for a unit change in GRANTING ACCESS or DEVOLVING CONTROL). In the case of handheld computers, the coefficients were statistically significant at the at least the 5% level (Models 5-1 and 5-2). Statistical significance was appreciably lower in the smartphone submarket with far fewer degrees of freedom and less variation; dropping time trends in this case increase statistical significance to 10% levels.\(^{43}\)

**Openness Increases the Scope for Diversity** Another building block in reduced-form hypotheses developed in Section 2 was the assumption that openness increases “diversity benefits” by exposing more pieces of a technology to differentiated contributions by heterogeneous suppliers. It is inherently difficult to distinguish the extent to which an open technology allows variegated contributions versus leads to them endogenously. Nonetheless, as suggestive evidence of such an effect, a measure of variation of products is regressed on measures of openness. Variation is measured as the ratio of differentiated products to total products launched. Differentiated products are identified as those products possessing any novel feature or level of performance in any of the 58 measured product dimensions. The ratio is based on accumulated

\(^{42}\)An annual change, or change over four quarters was used despite a quarterly data set given it is a relatively slow-moving variable.

\(^{43}\)Signs and magnitude of these coefficients remained intact even when applying stringent year fixed effects controls (not reported).
### TABLE 5: Within-System Entry Regressed on System Openness

<table>
<thead>
<tr>
<th>Within-System Entry</th>
<th>Handhelds Computers</th>
<th>Smartphones</th>
<th>Smartphones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(N=104)</td>
<td>(N=55)</td>
<td></td>
</tr>
<tr>
<td>Dependent Variable:</td>
<td>Crowding</td>
<td>Δ Suppliers</td>
<td>Δ Suppliers</td>
</tr>
<tr>
<td></td>
<td>[suppliers / units]</td>
<td>(year)</td>
<td>(year)</td>
</tr>
<tr>
<td>Model:</td>
<td>(5-1)</td>
<td>(5-2)</td>
<td>(5-3)</td>
</tr>
<tr>
<td>Specification</td>
<td>FE/OLS</td>
<td>FE/OLS</td>
<td>FE/OLS</td>
</tr>
<tr>
<td>Openness Variables</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Granting access</td>
<td>0.002</td>
<td>1.2</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>0.0007*</td>
<td>0.5657*</td>
<td>2.519</td>
</tr>
<tr>
<td>Devolving control(1)</td>
<td>0.002</td>
<td>1.2</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>0.0004**</td>
<td>0.4249**</td>
<td>1.6204</td>
</tr>
<tr>
<td>Time</td>
<td>0.0007*</td>
<td>.71</td>
<td>5.1716**</td>
</tr>
<tr>
<td></td>
<td>(.00)</td>
<td>(.00)</td>
<td>(.958)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(.13)</td>
</tr>
<tr>
<td>Time2</td>
<td>0</td>
<td>0.008</td>
<td>0.0366**</td>
</tr>
<tr>
<td>R²</td>
<td>0.85</td>
<td>0.57</td>
<td>0.78</td>
</tr>
</tbody>
</table>

Huber-White robust standard errors in parentheses

* significant at 5%; ** significant at 1%

(1) measured by count of number of component under platform supplier control.

(2) Includes (lagged) units sold at the level of the system, the submarket within a given system, the total submarket and the total industry, along with quadratic and growth terms.

Figure 3-8: Within-System Entry Regressed on System Openness
Regression controls in this case, $X$, include number of products (both cumulative and new releases), number of suppliers, time trends for each system, and measures of upgrades of the platform itself, which might lead to greater scope for diverse products.

\[
\frac{\text{NOVEL PRODUCTS}}{\text{PRODUCTS}}_{it} = \beta_A(\Theta) \cdot \text{ACCESS}_{it} + \beta_C(\Theta) \cdot \text{CONTROL}_{it} + \beta X_{imt} + \delta_{it} + \eta_{im} + \epsilon_{imt}
\]

(3.4)

Regression results are reported in Table 7. Regressions were performed separately for handheld computers and smartphones. Over a range of different combinations of control variables (Models 7-2 to 7-4) regressions using data for handheld computers consistently find that DE-VOLVING CONTROL is associated with higher variety, whereas GRANTING ACCESS appears to have no affect. This is consistent with the principal effect of granting access to be to change the number of suppliers that enter into the system\(^{45}\); whereas the effect of devolving control may be more subtle, with an effect on entry as well as on the scope over which suppliers can differentiate. Results are not statistically significant in data in the smartphone submarket, possibly a result of the limited variation and degrees of freedom in this case.

Openness Affects Strategic Investment Incentives through its Effect on Within-System Market Structure \footnote{The stock rather than flow measure was used to avoid on the one hand, the high variation from period to period and on the other hand an approach of creating a moving average of arbitrary duration.} The crucial assumption underlying the possibility that greater openness can lead to a negative relationship with increasing openness is the idea that greater openness and entry eventually leads to competitive crowding and diminished investment incentives in the handheld submarket, but crowding is averted in the smartphone submarket where industry structure leads to more powerful strategic complementarities. To test this assumption, individual firm level product investment decisions are analyzed by regressing new product releases on the extent of entry in the system. The basic model is similar in structure to the earlier models of low-coordination innovations, exploiting the same control variables and the

\footnote{A component of the score used to tally GRANTING ACCESS was intended to capture the scope allowed for differentiation through changes in the technology, itself (apart from changing control), but this component is not significantly correlated with the variation measure.}
panel structure to attempt to isolate exogenous variation in the key explanatory variables to identify the relationship. However, time trend variables are replaced with year and quarter dummies, given the wider cross-section should allows these to be estimated without bias and these should offer more stringent control of omitted variables (and possible selection-related biases related to when firms choose to enter). Fixed effects in this model are defined according to each individual firm in each individual platform, so device suppliers producing products on more than one platform (“multi-homing”) will have their product releases on each platform modeled separately.\(^{46}\) Supplier-level control variables, \(W\), were also added to regressions.

\[
\text{NEW PRODUCT RELEASES}_{ijt} = \kappa (\Theta) \text{ENTRY}_{it} + \kappa_2 (\Theta) \text{ENTRY}^2_{it} + \beta_X X_{imt} + \beta_W W_{imt} + \delta_{mt} + \alpha_{ijm(-t)} + \alpha_{ilm(-t)} + \gamma N_{i(-t)} + \eta_{ilm} + \epsilon_{imt}
\]  

(3.5)

The results, presented in Table 6, are consistent with crowding-out of investment in the handheld computer submarket but not in the smartphone submarket, as conjectured. In handheld computers, the number of new products developed by handheld computer suppliers increased with the number of suppliers with low numbers of suppliers, but the relationship turns negative at high numbers of suppliers, in an inverted-U pattern (Model 6-1). The average first order relationship over the sample was zero (not reported). This is consistent with crowding eventually overwhelming any stimulating effects of entry. Replacing NO. SUPPLIERS with a more direct measure of CROWDING—the number of suppliers, divided by unit sales—reveals a similar “inverted-U” pattern (Model 6-2), with a slightly better overall model fit (log-likelihood decreases from highly significant -684 to very slightly more significant -688).\(^{47}\) In the smartphone submarket, there is no such evidence of firms releasing fewer products with higher entry; product releases were positively related to the number of suppliers (6-4) and not related to the

\(^{46}\)This is a minority of firms. The results are insensitive to alternative approaches to modeling this problem, or even the exclusion of these firms.

\(^{47}\)Consistent with the ratio of crowding being perhaps less sensitive than absolute counts of suppliers to endogeneity problems, the inverted-U relationship with crowding has a less pronounced upward component and more pronounced negative slope in its “inverted-U” shape. A log-linear instrumental variables specification with crowding variables instrumented with the system-level openness variables produces a more negative slope, still (not reported).
direct measure of crowding at all (6-5), consistent with early bandwagon effects and virtuous circles between entry and investment. Regressions were also run using a standard dynamic panel specification (Arellano and Bond 1991) to assure similar results (Models 6-3 and 6-6), given the relatively short time series in the case of firm-level data, in conjunction with the use of lagged dependent variables risks introducing correlation between the lagged variables and the error term.

Crucial to the interpretation here of openness being used to regulate within-system entry to impact innovation, is that within-system entry must have a distinct effect apart from entry in the industry as a whole. Equivalently, strategic interactions—be they complementary or substitutive—should be more pronounced within a system, rather between suppliers in different systems. Consistent with this conjecture, measures of total numbers of firms in the industry (Models 6-1 and 6-4) or crowding measure for the entire industry (Models 6-2 and 6-5) are not significantly correlated with individual product release decisions.48

3.8 Summary and Conclusions

Should an innovator open its technology system to increase the system’s innovativeness and therefore its competitiveness against other systems? As an essential ingredient to answering this question, this paper empirically studied the relationship between degree of openness implemented by innovators and the amount of innovation that resulted. The question was investigated by analyzing panel data on competing systems in the mobile computing industry. Measures of innovation were developed on the basis of the changing features and performance of products built on top of the platforms in each of several leading systems. Measures of open systems policies were categorized as either relating to devolving control over a system by the platform owner or otherwise the granting of access to the platform by the platform owner.

The literature review suggests that a wide range of factors will mediate the openness-innovation relationship. However, it is argued that the nature of innovation and the nature of strategic interactions between suppliers will be particularly crucial determinants of the shape and magnitude of the relationship. In cases where there are very high complementarities be-

---

48 Similar results are found when including just the firms of the largest systems, or including firms within the same submarket only.
TABLE 6: Individual Supplier Product Development Decisions Regressed on Within-System Market Structure

<table>
<thead>
<tr>
<th>Dependent Variable: New Product Releases by Individual Firms</th>
<th>Handholds Computers (N=1244; 102 firms)</th>
<th>Smartphones (N=483; 55 firms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model: (6-1) (6-2) (6-3)</td>
<td>(6-4) (6-5) (6-6)</td>
<td></td>
</tr>
<tr>
<td>Specification FE/NB FE/NB Arr-Bnd(2)</td>
<td>FE/NB FE/NB Arr-Bnd</td>
<td></td>
</tr>
<tr>
<td>Within-System Market Structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. suppliers</td>
<td>0.40** (1)</td>
<td>0.64* (3)</td>
</tr>
<tr>
<td></td>
<td>(0.1)</td>
<td>(0.04)</td>
</tr>
<tr>
<td>No. suppliers2</td>
<td>-0.06** (0.002)</td>
<td>-0.015 (0.010)</td>
</tr>
<tr>
<td></td>
<td>(0.002)</td>
<td>(0.001)</td>
</tr>
<tr>
<td>No. suppliers, total industry</td>
<td>0.005 (0.045)</td>
<td>0.16 (0.16)</td>
</tr>
<tr>
<td></td>
<td>(0.005)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>Crowding [suppliers/units]</td>
<td>207** (62)</td>
<td>-3 (34)</td>
</tr>
<tr>
<td></td>
<td>(58 (53)</td>
<td></td>
</tr>
<tr>
<td>Crowding2</td>
<td>-4756** (1468)</td>
<td>16 (126)</td>
</tr>
<tr>
<td></td>
<td>(-1742 (1166)</td>
<td></td>
</tr>
<tr>
<td>Crowding, total industry</td>
<td>115 (201)</td>
<td>2630 (2258)</td>
</tr>
<tr>
<td></td>
<td>(-53 (92)</td>
<td></td>
</tr>
<tr>
<td>Firm Control Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New entrant dummy</td>
<td>2.1** (0.2)</td>
<td>1.5** (1.3)</td>
</tr>
<tr>
<td></td>
<td>(2.0** (0.1)</td>
<td>(1.7** (0.3)</td>
</tr>
<tr>
<td>Platform owner dummy</td>
<td>1.4* (0.6)</td>
<td>0.16 (0.16)</td>
</tr>
<tr>
<td></td>
<td>(1.3* (0.3)</td>
<td>(0.8** (0.1)</td>
</tr>
<tr>
<td>Percent ownership</td>
<td>0.01 (0.01)</td>
<td>0.00 (0.01)</td>
</tr>
<tr>
<td></td>
<td>(0.02 (0.01)</td>
<td>(0.01)</td>
</tr>
<tr>
<td>Multi-homing</td>
<td>-0.47 (0.38)</td>
<td>-0.52 (0.31)</td>
</tr>
<tr>
<td></td>
<td>(-0.48 (0.22)</td>
<td>(-0.81 (0.84)</td>
</tr>
<tr>
<td>Time since entry</td>
<td>-0.02 (0.02)</td>
<td>-0.23 (0.25)</td>
</tr>
<tr>
<td></td>
<td>(-0.03 (0.09)</td>
<td>(-0.25 (0.61)</td>
</tr>
<tr>
<td>Dependent variable, lagged</td>
<td>-0.01 (0.05)</td>
<td>-0.33* (0.14)</td>
</tr>
<tr>
<td></td>
<td>(-0.02 (0.04)</td>
<td>(-0.14 (0.14)</td>
</tr>
<tr>
<td>System Control Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. products, lagged</td>
<td>-0.02 (0.01)</td>
<td>-0.18 (0.13)</td>
</tr>
<tr>
<td></td>
<td>(-0.02 (0.01)</td>
<td>(-0.08 (0.09)</td>
</tr>
<tr>
<td>No. products, cum., lagged</td>
<td>0.00 (0.01)</td>
<td>0.08 (0.07)</td>
</tr>
<tr>
<td></td>
<td>(0.00 (0.01)</td>
<td>(0.03 (0.05)</td>
</tr>
<tr>
<td>No. suppliers, lagged</td>
<td>-0.16** (0.06)</td>
<td>0.35 (0.26)</td>
</tr>
<tr>
<td></td>
<td>(-0.01 (0.04)</td>
<td>(0.04 (0.15)</td>
</tr>
<tr>
<td>Other Control Variables</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform-firm FE</td>
<td>Sign. Sign. na/a</td>
<td>Sign. Sign. na/a</td>
</tr>
<tr>
<td>Log-Likelihood</td>
<td>-684 -688</td>
<td>-200 -202</td>
</tr>
</tbody>
</table>

Standard errors in parentheses
* significant at 5%; ** significant at 1%

(1) Includes (lagged) units sold at the level of the system, the submarket within a given system, the total submarket and the total industry, along with quadratic and growth terms. Includes dummy for smartphone market in cases where data from each submarket

(2) Arellano-Bond dynamic panel specification. Differences equation with difference of lag of dependent variable instrumented with second lag of the absolute level of the dependent variable

(3) The number of control variables is reduced in the A-B specification using the smartphone data, as described in the main text.
TABLE 7: Within-System Variation of Products Regressed on Openness

<table>
<thead>
<tr>
<th>Dependent Variable: Product Variation [Novel Products / Total Products]</th>
<th>Handhelds Computers (N=104)</th>
<th>Smartphones (N=55)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model:</td>
<td>Specification</td>
<td>FE/OLS</td>
</tr>
<tr>
<td>(7-1)</td>
<td>(7-2)</td>
<td>(7-3)</td>
</tr>
<tr>
<td><strong>Openness Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Granting access</td>
<td>.002</td>
<td>.004</td>
</tr>
<tr>
<td></td>
<td>(.005)</td>
<td>(.004)</td>
</tr>
<tr>
<td>Devolving control</td>
<td>.013**</td>
<td>.013**</td>
</tr>
<tr>
<td></td>
<td>(.004)</td>
<td>(.004)</td>
</tr>
<tr>
<td><strong>Control Variables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product releases</td>
<td>.004</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td>(.003)</td>
<td>(.003)</td>
</tr>
<tr>
<td>Product releases²</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td>(.000)</td>
<td>(.000)</td>
</tr>
<tr>
<td>Product releases, cum</td>
<td>.001*</td>
<td>-.002</td>
</tr>
<tr>
<td></td>
<td>(.000)</td>
<td>(.001)</td>
</tr>
<tr>
<td>Product releases, cum²</td>
<td>-6E-6**</td>
<td>(.000)</td>
</tr>
<tr>
<td></td>
<td>(1E-6)</td>
<td>(.000)</td>
</tr>
<tr>
<td>Platform age</td>
<td>-.10**</td>
<td>.17*</td>
</tr>
<tr>
<td></td>
<td>(.03)</td>
<td>(.08)</td>
</tr>
<tr>
<td>Platform version age</td>
<td>.083**</td>
<td>-.18*</td>
</tr>
<tr>
<td></td>
<td>(.03)</td>
<td>(.08)</td>
</tr>
<tr>
<td>Platform version launch date</td>
<td>.08**</td>
<td>-.18*</td>
</tr>
<tr>
<td></td>
<td>(.03)</td>
<td>(.08)</td>
</tr>
<tr>
<td>Platform FE</td>
<td>Sign.</td>
<td></td>
</tr>
<tr>
<td>Platform version FE</td>
<td>Sign.</td>
<td></td>
</tr>
<tr>
<td>R²</td>
<td>.84</td>
<td>.92</td>
</tr>
</tbody>
</table>

Huber-White robust standard errors in parentheses
* significant at 5%; ** significant at 1%

Figure 3-10: Within-System Variation of Products Regressed on Openness
tween entry and investment decisions—such as in cases of strong bandwagon effects—openness was predicted to lead to unconditional increases in innovation. In cases of lower complementarities, it was predicted that increased competitive crowding and rent dissipation will lead to lower innovation at high levels of openness and entry. In contrast, high-coordination innovations (also corresponding with “architectural,” “systemic” or “core” innovations in this case) are expected to be shaped by an ambivalent response of investment incentives to greater openness and diminished efficacy of the coordination regime around an open technology.

Following these arguments, the sample was divided into low coordination and high-coordination innovations and further divided into handheld and smartphone submarket (low complementarity versus high complementarity environments). The essential econometric exercise is to regress the different types of innovation on measures of innovation, while controlling for alternative determinants of innovation outcomes and determining whether the coefficients across contexts of high and low coordination and high and low complementarities reflect the predictions. There are two main identification challenges in this exercise. The first is a problem of omitted variables, where unobservable factors correlated with both openness variables and innovation outcomes lead to biases in the measured coefficients. In the absence of a clear natural experiment or instrumental variables, the approach taken is to exploit the panel structure of the data set and to apply numerous controls to attend to unobserved variation which might otherwise lead to bias. The second identification challenge relates to the fact that the observed slope of the openness-innovation curve will depend on the “part of the curve” being observed. This problem is addressed by interpreting coefficients accordingly, including second-order terms in regressions, and by studying the deeper economic mechanisms affecting the openness-innovation relationship to ensure proper interpretations.

The results were consistent with predictions and confirm an acute relationship between openness and innovation. The nature of innovation was found to be a key determinant of how innovation responded to openness. Low-coordination innovations closely were found to be closely related to the extent to which control was devolved over the system and outside device suppliers were allowed to enter into production in the system. These innovations appeared to largely be generated by outside suppliers building products “on top” of platforms, and followed the predicted patterns over multiple regressions involving different measures of innovation. It
was only in the case of both high complementarities (smartphones) and low coordination problems (modular innovation) that greater openness unconditionally led to increases in innovation. In a submarket with lower complementarities acting between suppliers (handheld computers), the multiple measures of low-coordination innovations first increased with greater openness, but eventually decreased—in an “inverted-U” pattern.

The economic and strategic importance of these effects was unmistakable across multiple measures of innovation, multiple regressions, multiple specifications and robustness checks and multiple levels of analysis. Variation in the measure of devolving control by one standard deviation led to a one-third increase in the generation of modular innovations in smartphones by roughly one-half, per period. Variation in the measure of the degree to which entry by device suppliers is promoted into a system (“access,” herein) more directly results in a multiplication of the number of suppliers in a system; thus variation of this measure by one standard deviation resulted in the measures of innovation to multiply by a factor that ranged between two-times and nine-times, depending on the particular measure of innovation.

However, the effect of increasing openness in handheld computers, where complementarities are weaker, illustrates a potential hazard of opening “too far.” At low levels of openness, increasing openness can increase low-coordination innovations with similar multipliers, but at high levels of innovation the marginal effect becomes sharply negative. The impact of one standard deviation variation in greater access to the system at high levels of openness (but still within sample) reverses to -60%. These relationships are potent and dramatic in their effects. They vary crucially on the nature of strategic interactions and the degree of openness in question. Later examination in the paper of the underlying economic mechanisms shows these pronounced effects are created by the means through which the degree of openness shapes market structure within the system, and resulting entry and investment patterns.

Consistent with predictions, the relationship between openness and multiple measures of high-coordination innovations was found to be far more ambivalent. These innovations were largely associated with core technologies in the platform, itself, and generated largely but not exclusively by the original innovator-cum-platform supplier. The magnitude of multiple measures of innovation were found to irregularly relate to the degree of openness—and mostly to not relate at all. Where the magnitude of advances did not appear to systematically relate to
the degree of openness, greater degrees of openness increased the frequency of these innovations, consistent with the increased involvement of outside suppliers in these innovations with greater openness. The flat response of the magnitude despite this increased frequency and weak negative correlations between magnitude and frequency of innovations suggest a potential trade-off with greater openness.

The Degree of Openness as a Strategic Instrument  So how open should a platform be made? It appears that in addition to the long-studied appropriability-adoption trade-off, trade-offs in relation to the effect on changes in the technology, itself, should be considered by the innovator of a technology. The analysis suggests the list of concerns facing the innovator should expand from a trade-off between “appropriability and adoption” to include “coordination, incentives and diversity”—insofar as technology racing between technology systems is an important mode of competition.

A central implication of these findings is that the trade-offs with coordination, incentives and diversity suggest a far more nuanced set of issues where polar cases of “fully opened” or “fully closed” may not be optimal in any sense, let alone profit-maximizing for the innovator. The empirical analysis confirms, as well, that the response of openness to innovation and therefore any optimal setting of openness will depend on the nature of the innovation task in question and the nature of strategic interactions of entering suppliers. For the most part, these may be exogenous features of industry and technology structure. Insofar as greater degrees of control and allowing or foreclosing of access favors different types of innovation, the particular design path chosen by the innovator might optimally be decided simultaneously with the openness decision. Very roughly speaking the results appear to confirm the suspicion of many practitioners and the suggestions of numerous scholars that “integrated innovation” may favor architectural, systemic, high-coordination innovations, of core subsystems; and “open innovation” may favor the autonomous development of modular innovations in peripheral subsystems.

Notwithstanding important distinctions and patterns that could be discerned in this exercise, the basic empirical facts of this industry—including central, powerful platform suppliers who “regulate” access to the system and govern conduct in the system more generally—suggest that far deeper and elemental issues of contracting, institutions, strategic technology and market 123
interactions are at work. These beg closer empirical study and formal characterization.
Bibliography


3.9 Appendix A: Reduced-Form Characterization of Response to Exogenous Changes in Openness

The following discussion offers a simple reduced-form characterization of the arguments made in the main text. Given the greater ability of the original innovator (who retains control over a critical subset of the technology) to implement innovations involving difficult coordination problems, and given the greater ability of multiple outside suppliers to foment bandwagon effects and provide diversity benefits, I will assume a priori that high-coordination innovations are implemented by the original innovation and low-coordination innovations are implemented by the entering suppliers. To reflect the research objective and empirical strategy, I focus here on responses to exogenous shocks in openness.

Openness and Low-Coordination Innovations by Entering Suppliers Let the reduced-form relationship between openness and low-coordination innovations by entering suppliers be given by the following expression:

\[ Y_{LC} = g(O, \Theta) \]

where \( Y \), the total innovation generated by the entering suppliers, is presumed to be affected by openness of the system, \( O \). In this paper, openness has been defined as granting access and devolving control. (These will be considered jointly as \( O \) to the extent they imply the same predictions.) The openness-innovation relationship is mediated by industry structural features, \( \Theta \). The function \( g \) therefore captures in reduced form all behavioral relationships relating innovation to openness, as mediated by industry structure. Assume the expression can expressed the multiple of the number of outside suppliers entering to the system, \( n \), times the average innovation output of each entering supplier, \( y \), where each of these arguments may take the same arguments as the earlier expression, or, \( Y = n \cdot y \). The slope of the openness-innovation relationship might then be understood in terms of the total derivative of the earlier expression, without unduly assuming the structure of key economic relationships.

\[ \text{\footnotesize\textsuperscript{49}} \text{The exposition focuses on within-system changes with openness, rather than interactions with other systems.} \]
The above expression roughly organizes the points made in the main discussion. The effect of increasing low-coordination openness on innovation as implied by the first two terms is the diversity effect: greater openness increases the number of suppliers, whose direct effect will be positive; greater openness may also imply a direct increase in the scope for differentiation and productivity of each individual supplier. The final term is the effect of openness on individual innovation as mediated by entry into the system, or the strategic incentives effect. This may be positive or negative depending on a host of issues considered in the discussion. The overall expression clarifies that the slope of the openness-innovation relationship cannot be known a priori, but will depend on industry structure. Further the slope should also be a function of the level of openness, insofar as (non-negative) diversity effects vary in magnitude, possibly with diminishing returns, and strategic incentives effects will also surely change with greater openness, entry and possibly greater intensity of competition.50

### Openness and High-Coordination Innovations by the Original Innovator

The effect of greater openness on high coordination innovations will be the sum of strategic incentives effects and coordination effects, if these innovations are implemented by the original innovator (and without a plurality of contributors there is no diversity effect). Coordination effects are assumed to be negative outright, based on the findings of the previous literature, and I focus here on strategic incentive effects of greater openness on high-coordination innovations.

The optimizing level of high coordination innovation by the original innovator will depend upon the share of total surplus in the system that can be captured by the original innovator, times the total surplus that can be created in the system, which is assumed at this point to solely relate to investments in product innovation by entering suppliers, as earlier given, and investments by the original innovator in high coordination innovations, $Y_{HC}$. Openness is treated as a environmental parameter given we focus here on exogenous shocks.

---

50 Levels of investment may also change with changing investment of complementary investments by the original innovator; however, the following discussion argues that the response of the original innovator’s investment to openness will be ambivalent and so this effect will likely be of a lower order and therefore is ignored here.
\[ Y_{hc}^* = \arg \max_{\{y_{hc}\}} \Pi(\sigma, y_{hc}, y_{lc}; o, \theta) \]

The marginal return to the original innovator’s own innovation will be given by the following expression. I assume that the effect of investment on bargaining power are of lower order than the direct effect of investment on surplus creation.

\[
MR \equiv \frac{d\Pi}{dY_{hc}} = \frac{\partial \Pi}{\partial Y_{hc}} + \frac{\partial \Pi}{\partial \sigma} \frac{\partial \sigma}{\partial Y_{hc}} + \frac{\partial \Pi}{\partial Y_{lc}} \frac{\partial Y_{lc}}{\partial Y_{hc}} \approx \frac{\partial \Pi}{\partial Y_{hc}} + \frac{\partial \Pi}{\partial Y_{lc}} \frac{\partial Y_{lc}}{\partial Y_{hc}}
\]

To consider the innovation response of the original innovator of the system to greater openness, I consider the effect of greater openness on the marginal return to investment. I assume the strategic effect of the original innovators investment on the outside device suppliers’ investments (second term in the preceding expression) does not change appreciably with a marginal increase in openness, giving the following expression:

\[
\frac{dMR}{dO} \approx \frac{\partial \Pi_{Y_{hc}}}{\partial Y_{hc}} \frac{\partial Y_{hc}}{\partial O} + \frac{\partial \Pi_{Y_{hc}}}{\partial \sigma} \frac{\partial \sigma}{\partial O} + \frac{\partial \Pi_{Y_{hc}}}{\partial Y_{lc}} \frac{\partial Y_{lc}}{\partial O}
\]

If greater openness will tend to diminish the productivity of the innovator’s investments by exacerbating coordination problems, the direct effect of greater openness will also be to diminish the innovator’s investments. The bargaining effect may be positive or negative (granting access will increase the innovator’s bargaining power in relation to entering suppliers; devolving control will work in the opposite direction), but will generally work in the opposite direction of the correlated investments effect, because greater bargaining power for the innovator will mean lower bargaining power and incentives of the entering suppliers.

It is not possible to sign this expression without many more structural details, but the effect of openness on the marginal returns of investments should be relatively ambivalent because of the negative correlation of bargaining and correlated investments effects. These incentive effects, plus a possibly generally negative effect of openness on innovation will produce an ambivalent and possibly negative relationship between low-coordination innovations and openness.

3.10 Appendix B: Variables
<table>
<thead>
<tr>
<th>Discrete Measures of Performance</th>
<th>Continuous Measures of Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>device type</td>
<td>internal camera</td>
</tr>
<tr>
<td>pda</td>
<td>microphone</td>
</tr>
<tr>
<td>phone</td>
<td>features</td>
</tr>
<tr>
<td>dataterminal</td>
<td>hard drive</td>
</tr>
<tr>
<td>twoway pager</td>
<td>backlight</td>
</tr>
<tr>
<td>form factor</td>
<td>recharge battery</td>
</tr>
<tr>
<td>penpad</td>
<td>infrared</td>
</tr>
<tr>
<td>watch</td>
<td>Comms</td>
</tr>
<tr>
<td>clamshell</td>
<td>bluetooth</td>
</tr>
<tr>
<td>slate tablet</td>
<td>cdma</td>
</tr>
<tr>
<td>subnotebook</td>
<td>gsm</td>
</tr>
<tr>
<td>brickphone</td>
<td>mobitex</td>
</tr>
<tr>
<td>flipphone</td>
<td>cdma</td>
</tr>
<tr>
<td>watch</td>
<td>reflex paging</td>
</tr>
<tr>
<td>landscape</td>
<td>wifi</td>
</tr>
<tr>
<td>application</td>
<td>internal modem</td>
</tr>
<tr>
<td>industrial</td>
<td>port slots</td>
</tr>
<tr>
<td>real estate</td>
<td>usb</td>
</tr>
<tr>
<td>medicine</td>
<td>serial</td>
</tr>
<tr>
<td>mediaterminal</td>
<td>parallel</td>
</tr>
<tr>
<td>videogames</td>
<td>cf slot</td>
</tr>
<tr>
<td>audioplayer</td>
<td>sds</td>
</tr>
<tr>
<td>design input</td>
<td>memory stick</td>
</tr>
<tr>
<td>color fashion case</td>
<td>others slots</td>
</tr>
<tr>
<td>metal case</td>
<td></td>
</tr>
<tr>
<td>plastic case</td>
<td></td>
</tr>
<tr>
<td>thumb board</td>
<td></td>
</tr>
<tr>
<td>full keyboard</td>
<td></td>
</tr>
<tr>
<td>touchscreen</td>
<td></td>
</tr>
<tr>
<td>hw recognition</td>
<td></td>
</tr>
<tr>
<td>jog dial</td>
<td></td>
</tr>
<tr>
<td>rugged</td>
<td></td>
</tr>
<tr>
<td>twist screen</td>
<td></td>
</tr>
<tr>
<td>toggle nav with thumb</td>
<td></td>
</tr>
<tr>
<td>hardware</td>
<td></td>
</tr>
<tr>
<td>barcode</td>
<td></td>
</tr>
<tr>
<td>gps</td>
<td></td>
</tr>
<tr>
<td>processor speed</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>screen resolution</td>
<td></td>
</tr>
<tr>
<td>color depth</td>
<td></td>
</tr>
<tr>
<td>physical size</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3-11: List of Product Characteristics Measured
<table>
<thead>
<tr>
<th>SYSTEM COMPONENTS</th>
<th>SUB-COMPONENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATIONS</td>
<td>Applications</td>
</tr>
<tr>
<td></td>
<td>Specialized, high-level programs used directly by the end user</td>
</tr>
<tr>
<td></td>
<td>Integ'd Dev't Env't (IDE) &amp; Tools</td>
</tr>
<tr>
<td></td>
<td>Software tools developed for use by programmers, incorporating the basic frameworks, intended to facilitate development of application</td>
</tr>
<tr>
<td></td>
<td>Framework &amp; App'n Prog'g I/F (APIs)</td>
</tr>
<tr>
<td></td>
<td>Definition of applications programming interfaces and other basic rules programmers need follow</td>
</tr>
<tr>
<td>OPERATING SYSTEM</td>
<td>Graphical User Interface (GUI)</td>
</tr>
<tr>
<td></td>
<td>Software-defined visual display of system</td>
</tr>
<tr>
<td></td>
<td>Utilities</td>
</tr>
<tr>
<td></td>
<td>Memory resident program that performs a very specific task, usually related to managing system resources</td>
</tr>
<tr>
<td></td>
<td>Kernel</td>
</tr>
<tr>
<td></td>
<td>Translates high-level instructions from applications programs and input-output devices to machine-level instructions</td>
</tr>
<tr>
<td>DEVICE</td>
<td>Collection of board-level design, system integration, industrial design tasks that yields a working device</td>
</tr>
<tr>
<td></td>
<td>High-level definition</td>
</tr>
<tr>
<td></td>
<td>System bus design, processor compatibility</td>
</tr>
<tr>
<td></td>
<td>Board-level Specs</td>
</tr>
<tr>
<td></td>
<td>Physical attributes, component selection</td>
</tr>
<tr>
<td></td>
<td>Complete design and integration</td>
</tr>
<tr>
<td></td>
<td>Working, integrated, tested designs</td>
</tr>
<tr>
<td>PROCESSOR</td>
<td>(Set of) component(s) designed to perform basic arithmetic and logic operations As defined here, also radio signal processing, low-level control of basic components Receives instructions from the operating system</td>
</tr>
<tr>
<td></td>
<td>Drivers and configuration</td>
</tr>
<tr>
<td></td>
<td>Low level code embedded on processor, itself, to carry out basic machine-level operations</td>
</tr>
<tr>
<td></td>
<td>Microprocessor</td>
</tr>
<tr>
<td></td>
<td>Integrated circuit device</td>
</tr>
</tbody>
</table>

Figure 3-12: Characterization of the Components in Systems
access_licensing_sum
access_single_supplier
access_restricted_licensing_per_niche
access_restricted_licensing_in_aggregate
access_limited_licensing_capacity
access_unlimited_licensing

access_entry_sum
access_detailedhwspecs
access_hw_development_support
access_ref_design

access_diff_sum
access_mutliple_diffld_platforms
access_high_configurability
access_unfettered_configurability
access_allowed_to_modify_platform

Figure 3-13: Variables used to Construct Access Measure
Chapter 4

Schumpeterian versus Network Effects in Systems Industries

4.1 Introduction

The relationship between competition and innovation has come under sustained study for over a half-century (Schumpeter 1942; Arrow 1962; Dasgupta and Stiglitz 1980; Reinganum 1989; Aghion and Howitt 1992; Aghion et al 2005). Although much controversy remains in this research, theoretical and empirical research in this “neo-Schumpeterian” tradition now sheds light on how intensified competition may alternately increase or decrease innovation and, conversely, how innovation might also shape market structure. However, a vast and vital part of the economy in which innovation plays a vital role—system industries (those where products are made up of multiple components such as aerospace, telecommunications and IT)—remain largely unexplored in this work. While there have long been calls for understanding technical change in systems\(^1\) the special circumstances of these industries—including a pronounced role of network effects, technical standards, modularity in design, and a great many firms often organizing around a single system—have led to rather specialized literatures to examine these issues instead (Katz and Shapiro 1994; Baldwin and Clark 2000; Tirole and Rochet 2003).

\(^1\)Professor J. Marschak called for greater attention to questions of innovation in systems industries in The Rate and Direction of Inventive Activity: Economic and Social Factors NBER 1962. He pointed to the relative importance of these industries in the economy and the often extraordinary scale of systems development projects.
Between neo-Schumpeterian and Systems perspectives has emerged a wide gulf on the question of whether or not competitive entry will promote innovation in a particular system component. The starting point in the neo-Schumpeterian tradition is how entry affects competitive intensity and the resulting strategic incentives to invest in innovation. While there remains much debate on the nature of these “Schumpeterian”\textsuperscript{2} effects\textsuperscript{3} there is general consensus within this perspective that very high levels of competitive entry will reduce appropriability, resulting in reduced incentives to innovate—and less innovation.

The research focused on systems industries generally takes as a starting point that high levels of entry into the supply of a components will lead to higher quality and variety of the components because of network effects. The idea of promoting entry has gained considerable currency; the tens of thousands of firms in the “business ecosystems” created around Microsoft (Iansiti and Levien 2004) and Intel platforms (Gawer and Cusumano 2002) are now widely viewed as worthy of emulation. The implied rule of thumb is: “he who has the largest ecosystem wins.” On the basis of this logic, systems innovators, platform suppliers often actively promote the entry of component suppliers to their systems. But are network effects really so compelling and systems industries really so exceptional? Might the entry of fantastic numbers of component suppliers at times not be productive?

This paper studies a textbook example\textsuperscript{4} of platform suppliers: the promotion of entry of software developers into mobile computing systems (handheld computers and smartphones). Platform suppliers in mobile computing actively promoted the entry of large numbers of applications software developers with the interest of fomenting (indirect) network effects whereby a wide variety of applications software would lead to greater adoption of the platform by end-users, and a larger installed-base of end-users would lead to the supply of more software. Software developer developers were encouraged to enter through a combination of approaches taken by platform suppliers: the creation of standardized applications programming interfaces,\textsuperscript{5} the

\textsuperscript{2}The classic works of Joseph Schumpeter touched on issues beyond competitive intensity and strategic incentives. This is a contemporary usage (Aghion, et al. 2005).

\textsuperscript{3}See Aghion and Griffith (2005).

\textsuperscript{4}Software developers are among the most cited examples of a source of network effects (Katz and Shapiro 1985; Farrell and Saloner 1986; Church and Gandal 1992; Armstrong 2002; Tirole and Rochet 2003; Hagiu 2004; Evans, et al. 2004).

\textsuperscript{5}Application programming interfaces (APIs) are a library of high-level instructions to invoke lower level commands by the operating system.
provision of developer kits and tools, the provision of documentation and training related to
programming, the provision of sample source code, the provision of direct subsidies and invest-
ments in cases, and through the design of the overall system architecture in the first place to
facilitate interaction with developers. The extent and quality of this developer support has
generally increased over time in each platform supplier, with growing internal resources and
staff devoted to developer support as a distinct business function in these firms. As a result
of these efforts, tens of thousands of applications developer firms have supplied software in the
industry. Consistent with the systems perspective and the objectives of platform suppliers, the
data presented in this paper and earlier research (McGahan et al. 1997; Nair, et al. 2004) make
clear that large developer networks corresponded with larger libraries of software applications.

Notwithstanding the great variety of software that was produced in each system, it remains
a question whether the quality and variety of software created represented a best possible
outcome. Would fewer applications developers actually have led to a more valuable offering
of software to end-users and therefore had a stronger complementary effect on the demand
for the platform? The Systems perspective suggests not. But the Schumpeterian perspective
suggests that lower levels of entry by competing software developers may have led to more
investment and better software. This paper presents a very preliminary descriptive analysis of
the structure and demand for and supply of applications software in leading mobile computing
systems to assess whether lower entry by applications developers in these systems might have
potentially led to a more valuable selection of software created. The analysis is mainly based on
a detailed point-of-sale database of the leading on-line retailer of third-party mobile computing
applications software from 1999 to 2004.

The analysis begins with a basic description of key facts, suggesting a role for investment
incentives. Consistent with prior studies, numerous features of the microdata are consistent
with indirect network effects operating in this industry. However, there were simultaneous
suggestions of a role for strategic investment incentives and Schumpeterian effects. A first
key observation is that the majority of product variety and selection was not generated by
incremental entry of additional developers, but by the discretionary decisions of individual
developers already in the market to release more titles. Over 80% of the titles released are
from developers already in the market rather than new entrants. This suggests that there is
tremendous scope for strategic incentives to play a role, as any factors—including competitive intensity—that influence the discretion taken by incumbent developers should profoundly affect the overall selection of titles in a system. That a large minority of the titles released by these incumbents represent upgrades and extension further emphasizes the role of investment in potentially affecting not only variety but also quality of titles.

Adding further indication of the potential role of competitive crowding and loss of appropriability is evidence of competition between software developers. There was considerable overlap in types of software released by different firms and price reductions with growing entry. Moreover, the evidence suggests that marginal entrants often supplied inferior quality titles, at lower prices. While these individual marginal entrants each took minimal share on their own, as a group they captured a substantial fraction of the market (developers lower than 1000th in rank capture roughly a tenth of market share within each of Microsoft and Palm systems, as a group). And therefore, the conditions of the industry appeared to allow for the possibility that greater entry led to more competition and a less valuable selection of application software.

So what was the net effect of adding marginal software developers to these networks? Does greater entry, at some point, reduce the quality or variety of a developer network? This preliminary investigation investigates a simpler and stronger point: that more software developers may lead to fewer titles altogether if greater entry reduces incentives to develop new products by incumbent developers. The question is investigated in a simple panel framework where the number of software titles releases in a system for a given quarter is regressed on the number of active sellers of software in that system. Fixed effects for time and systems offer very conservative controls in this analysis; as discussed in the text, unobserved factors affecting both numbers of firms and the new title releases should create a positive bias in the regressions. Despite the inherently positive bias in these regressions, using both parametric and nonparametric methods I find a negative relationship between the rate of software development and the increasing size of developer networks. The findings suggest that rather than simply a case of diminishing contributions made by marginal entrants, the net effect of growing networks was in fact been negative. I interpret these findings as suggesting the pronounced role that strategic investment incentives can play in component markets.

This paper proceeds in Section 2 by reviewing the literature on systems to argue that in
the context of network effects there is still scope for investment incentives to play a role—and that greater entry can potentially lead to lower innovation output. Section 3 provides a basic description of the empirical context and the data used in this paper. Section 4 provides a main descriptive analysis. Section 5 further discusses and interprets the findings. Section 6 concludes.

4.2 Why Promote the Entry of Component Suppliers to a System?

The treatment of entry into inter-related markets in systems industries has garnered considerable attention. In questions of antitrust (Bresnahan 2000) and regulation (Bessen et al. 2002), efforts to deter entry have been treated with suspicion. In questions of firm strategy, the orientation has been equally pro-entry in orientation. Prescriptions tend to emphasize the value of creating “business ecosystems” or “networks” of suppliers around a system (Gawer and Cusumano 2002; Iansiti and Levien 2004). From this perspective, parties such as system innovators or platform suppliers with the ability to appropriate surplus created in the system will have incentives to actively promote and subsidize entry into component markets (Armstrong 2002; Rochet and Tirole 2003; Hagiu 2004). The following discussion reviews the arguments—mostly relating to the operation of network effects—for why promoting entry can be productive, resulting in greater surplus created in a system. The discussion then turns to clarifying how these arguments still leave room for the possibility of diminished investment incentives as the size of the “network” of component suppliers grows.

Indirect Network Effects and Entry by Component Suppliers The view that entry into market for the individual components of a system should be allowed and even promoted beyond free entry levels has its roots in theories of indirect network effects. In markets with network effects, the benefit derived from buying a good increases with the number of adopters (Katz and Shapiro 1985, 1986). In the case of indirect network effects, the added benefit comes from supply-side changes brought about by a larger installed base of users. The working premise of research in this area is that indirect network effects can lead to some combination of greater availability, increased variety, lower price and greater quality of system components, in response
to a growing installed base of end-users. Katz and Shapiro (1985), for example, motivated their work on network effects by observing that a growing installed base of computers, media players, video games and the like should lead to greater variety in the supply of software and media used on these devices, drawn by a growing market opportunity. In reference to the example of the provision of post-sales service and repairs, it may simply be easier to find or obtain a complement when the installed base is larger. Farrell and Saloner (1985) motivate their work in part by suggesting that complementary goods become cheaper and more readily available with the extent of the market for them, with special reference to the examples of spare parts, service and software. Katz and Shapiro (1994) describe the possibility of underlying economies of scale in producing complements as a potential underlying mechanism. Increased experience and mounting investments in training in the technology and production are mentioned by Katz and Shapiro (1985) as another potential reason why the quality of supply of complements could also improve with the size of the installed base. Theoretical research on network effects that follows these seminal papers, along with more explicit models of indirect network effects (Church and Gandal, 1992; Chou and Shy, 1990) and multi-sided markets (Armstrong 2002; Tirole and Rochet 2003) provide similar motivations, and then typically proceed to study the implications of these effects within highly stylized models.6

Where the system innovator or platform supplier cannot itself deliver these benefits7, the question remains why might high levels of entry, typified by software developer “networks,” are needed to generate these benefits? The answer is not immediate; where these benefits are rooted in economies of scale of any sort,8 in fact the exact opposite conclusion should be drawn.

The basic argument suggested in the literature on network effects concerns the role of market power in network effects, as summarize here by Katz and Shapiro (1994): “These indirect network effects are perhaps easiest to see when many firms offer differentiated software... When

---

6The research often begins with highly stylized formulations. The network effects literature generally builds models in which the benefit enjoyed by a buyer demands on the number of other users, i.e. \( u = u(n) \). Models of multi-sided markets formulate benefits in relation to the number of other types of market participants, i.e. \( u_i = u(n_j) \). More explicit models of indirect network effects have tended to focus on the role of increasing product variety without explicit mention of the structure of supply behind increasing variety of products, i.e. \( u = u(N_{products}) \).

7For example, a platform supplier may not be able commit to not holding-up buyers on post-purchase sales of add-on components, or it may simply lack the experience and skills.

8Ex: Economics of scale in sales, synergies in product development scope, experience curves, mounting (endogenous) investments (Sutton 1998), etc.
differentiated software is supplied by many firms with low entry barriers, theoretical models of monopolistic competition (Salop, 1979; Dixit and Stiglitz, 1977; Spence, 1976) indicate that the variety of software may be greater, and the price of software less, the larger is the total demand for software (Church and Gandal, 1992; Chou and Shy, 1990).” And so, to the extent there is market power in the supply of components, any deadweight losses and possible lower level of variety supplied may become amplified by indirect network effects operating between component supply and end-user adoption. From this perspective then, entry is instrumental to creating competitive conditions.

An additional explanation for why greater growing entry is that component suppliers may be heterogeneous and have comparative advantages in producing differentiated products (Baldwin and Clark 2000; Katz and Farrell 2000; Gawer and Henderson 2006). In such a case, the number of entrants might rise beyond simply the number required to bring competitive conditions to the industry. In the extreme, the draw of Ricardian rents for firms with different experience and specialities could conceivably lead to as many suppliers as varieties of components, if each supplier has preferred costs or quality in producing a single variety. In the context of highly uncertain innovation, these comparative advantages may take the form of sunk investments in alternative design paths. Investments in distinct design paths whose value can only be known once market and technical uncertainty is resolved may produce ex post heterogeneity among suppliers (Nelson 1961; Nelson and Winter 1977; Langlois and Robertson 1992; Baldwin and Clark 2002).

**Investment and Innovation in a System: Schumpeterian Effects versus Network Effects?** In contrast to these Systems perspectives that are often premised on the benefits of entry, the more general “neo-Schumpeterian” tradition of research on the relationship between competition and innovation (Schumpeter 1942; Arrow 1982; Dasgupta and Stiglitz 1980; Reinganum 1989; Aghion and Howitt 1992; Aghion et al. 2005) argues that the effect of entry will depend on its effect on competitive intensity and resulting strategic investment incentives of firms. Although there remains much controversy about the precise effects of entry and competition on innovation (Aghion et al. 2005; Aghion and Griffith 2005; Vives 2005), there is general consensus that very high levels of entry and competition will stifle innovation through
lost appropriability and a resulting drop in incentives to invest in innovation.

The conditions for this sort of outcome may be at work even in systems industries, where indirect network effects are at work. Mounting evidence and theory suggest that entry can intensify competition and reduce the profits of component suppliers⁹ (Church and Gandal 1992; Economides 1996; Ellison and Fudenberg 2003; Rysman 2003; Hagiu 2004; Economides and Skrzypacz 2004). If competitive crowding, acting within a component market, were to overwhelm the market expansion created by indirect network effects working from component markets across to end-users, the Schumpeterian effect of reduced innovation might lead to reduced innovation with increasing entry and a growing network of component suppliers. Examining whether this sort of tension in fact emerges in an empirical context is the main objective of this paper.

4.3 “Networks” of Applications Software Developers in Mobile Computing

Handheld computers and smartphones, collectively “mobile computers,” are small, programmable computing devices, running on battery power for many hours and sometimes even days or weeks before recharging. They are meant to be pocketed and used as personal devices. The freedom of use created by their portability, mixed with the constraints created by smaller screen sizes and input methods to accommodate their size (such as pen entry, thumbboards or voice recognition) lead these devices to be used in somewhat different ways than their larger cousins, personal computers. A most common application is as personal information manager (PIM), including calendar schedule, address book and to-do lists. However, their range of use extends broadly to include diverse applications such as financial tools to video games to maps and databases. These devices are also intended to virtually “extend” the computer desktop, and are frequently being used to “synchronize” with software on the desktop. This allows users to read, edit and enter data that can later be updated with the desktop through simple linking of the device through serial port, USB port, infrared connection or over the Internet. That these devices are bought by individuals (rather than IT departments) and then used for both

⁹This is also the very premise of Katz and Shapiro’s (1994) argument, earlier.
personal and professional applications has led to greater diversity still in their use. Smartphones are essentially handheld computers that are able to communicate with commercial mobile radio (cellular) networks and thus also allow voice communications and data communications, and thus a potentially wider variety of networking applications. Smartphones and handheld computers remain largely kindred technology as they have in many cases been built on the same operating system platforms and have a good deal of technology in common.

Several of the most popular “workhorse” applications programs generally come pre-loaded on the devices, including PIM software, thousands of other titles are available that users typically purchase independent of the device. But thousands of other applications have historically been supplied by “third party” applications suppliers at arm’s length. These are typically downloaded to a personal computer and then “synchronized” to the mobile computer in a simple sequence of commands; sometimes just the touch of a button. While it is difficult to find overall market estimates, numerous industry managers estimated that a large minority—perhaps one in four or one in five—device owners downloaded third party software and, if so, downloaded multiple titles. Estimates by Nair et al. (2003), suggest that positive feedback between growing end-user demand with growing supply of software selection accounted for a large share of demand variation (roughly 20% of the log-odds ratio in a multinomial logit model).

![Number of Software Applications Available](image)

Figure 4-1: Growing Numbers of Applications Software Around the Leading Platforms
Mobile computers, and handheld computers in particular, exploded as a category in the mid and late 1990s, buoyed by the success of the wildly successful Palm Pilot, released by US Robotics in March 1996. The Palm Pilot was described at the time as the fastest selling consumer electronics device in history, adopted by over three million users by year-end 1998. The overall mobile computing industry has roughly doubled annually since that time, reaching just under 12 million units by 2000 and growing to tens of millions shortly thereafter.

The structure of this industry (and technology) was of competing platform with multiple devices built on these platforms by multiple device suppliers. Other firms entering to supply components in systems included chipmakers, applications software developers, peripheral hardware makers, trainers, technical consultants, and value-added resellers. Platform owners devoted significant attention to promoting entry of the different types of complementors into these multiple markets. Company press releases and executives used language to evoke a sense of the mutually reinforcing relationships between the multiple sides. For example, Palm referred to its “Palm Economy;” Symbian, its ”Symbian Ecosystem;” and MontaVista, its “Open Framework.” This promotion of entry was viewed as a central element of platform strategy, as suggested by the following quotation by Mark Bercow, the Head of Partnerships at 3Com’s Palm Computing division in an August 1998 press release and the following exhibit of Symbian, reflecting this view.

“Our solution provider community has always been a primary factor in the success of the Palm Computing platform, and has broadened the functionality of our products to help us reach a wide spectrum of customers.”

Among the multiple “sides” of the market that formed for complementary goods around platforms, platform suppliers were perhaps most aggressive in promoting the entry of software applications developers. Whereas it has been common for hardware suppliers to enter in the dozens and for a handful of chip manufacturers to enter into a given platform, software developers have entered into mobile computer platforms in the thousands and tends of thousands. In 2004, Palm reported it had 320,000 developers; Nokia reported it had roughly 600,000 devel-

---

10 Numerous smartphones were commercially released in the early 1990s, but only gained commercial success in the 2000s.

11 Average of estimates by IDC, Gartner and other sources.
Figure 4-2: “The Smartphone Market Virtuous Circle,” reproduced from Symbian’s (2006) How Smartphones Work Wiley Publishers

operators (of which some fraction were developing applications to run on Nokia-supplied graphical user interfaces that worked on top of the Symbian operating system); Microsoft reported the number of .NET developers (including those for mobile computing, servers and desktop) was roughly 6 million. These numbers are difficult to interpret or verify, but judging from observable counts of published software titles (ranging from thousands to tens of thousands for leading platforms during the period), there were tens of thousands of firms or individuals actively producing commercial software by the early 2000s. Where numbers are a little less ambiguous is in the number of applications released. The following figure shows estimates of the enormous numbers of applications released by the leading platforms in the industry.

With cumulative sales of roughly 80 million units for the entire industry between 1990 and 2003 and an installed base of adopters of roughly 25 million individuals—for the entire industry—this implied that there might be roughly one application for every 300 users, where these users were individual consumers and many of which may not buy any third-party applications. The promotion of such extreme levels of entry was directly related to the belief that the supply of applications software was crucial to stimulating network effects. The nature of these indirect effects as conceived by industry executives (and later estimated by Nair, et al. (2004)) were succinctly described by McGahan, et al. (1997) as follows:¹²

¹²More recent commentary by executives calls the strength of these network effects into question (Yoffie, et al 2005).
“PDAs will become increasingly valuable when other users adopt the product... The variety and range of available software will increase with the number of users. This response has been endemic in consumer electronics. Recording companies offered a greater variety of titles on compact discs as sales of hardware grew, for example. Similarly, software that enhance the utility of the a PDA will grow as the user base expands. Just as the greater variety of CD titles induces additional buyers to purchase the CD hardware, so more software will induce greater purchases of PDAs.”

Given the perceived importance of the indirect networks effects, the active promotion of large software developer networks also became orthodoxy in the industry. At Psion, The practice of opening applications software goes at least back to Psion’s original opening of development of applications software with its earliest models, by allowing programming and modular memory even in its earliest system released in 1984. Substantial efforts began by Psion in the early 1990s to attract third party developers to its platform, through developer contests, the development of new modular memory (that would later evolve into the standard compact flash memory cards), through the provision of database templates and access to the use of database code, and through offers of co-marketing. However, the economics of putting simple, sometimes trivial applications, on expensive memory cards to charge over $100 per title to a yet exceedingly installed base of users, and on a somewhat specialized development language proved not to be viable.

By the middle of the decade leading platform suppliers Palm and Microsoft placed application developers at the center of their strategies. Microsoft added considerable complexity and risk to the development of their Windows CE platform in forcing their designs to retain the applications programming interface (API) library to known by desktop programmers—this despite having to construct an entirely different code set from the Windows kernel being emulated. When the Windows CE platform was launched in 1996, Microsoft also did so with a view of “seeding” adoption of the platform by simultaneously announcing 80 developers for the platform. The company continued to upgrade tools and promote the software developer network with regular upgrades to already reasonably mature and sophisticated tools (drawing substantially from PCs) and later integrating development and functionality within the broader
.NET framework, intended to bring relatively seamless development and functionality and data interchange between multiple types of devices and computers.

Similarly, Palm launched its platform ready-made with an API set and collaborated with Metrowerks to provide a development environment. Enthusiasm for the platform was so great that even in the absence of ready-made tools that could be used in a PC environment (the original tools only worked on the Macintosh), an enthusiastic community created its own development tools, based on the source code and system documentation that Palm had provided. Dozens and then hundreds of titles quickly emerged and Palm quickly had the greatest number of titles.

This emphasis on promoting developer networks was roughly replicated by following entrants such as RIM and Mobilinux. However, given the difficult task of catching up with enormous libraries of applications of market leaders have also tended to more aggressively port their platforms to middleware to gain rapid access to a larger set of applications. For example, RIM, with relatively little third-party software for the device relies on generic mobile Java (J2ME) applications to run on top of the device operating system to gain access to a wider library of applications. Similarly, MontaVista runs Trolltech’s Qtopia as an interface on top of Mobilinux and can thus use applications that work with Mobilinux; Symbian runs Nokia’s Series 60 interface on top and thus benefits from Nokia’s large developer network. While such an approach does not always lead to an advantage (for example, most other platforms have already embedded Java runtime engines), such strategies mitigate the disadvantages of having a small set of platform-specific applications.

4.4 Data Set

The main data source used in this study is a proprietary and confidential point-of-sale database of a leading Internet retailer of mobile computing software applications. Internet retailers or “aggregators” of software have dominated distribution. No major brick and mortar computer or consumer electronics manufacturer carries or sells mobile computing software of any meaningful selection or volume. It is difficult to confirm market share of Internet retailers, but there is broad consensus this retailer is a market leader. Comparisons with estimates of total industry sales of
applications software suggest this retailer carries a large minority of all industry transactions, and possibly even a majority.\textsuperscript{13} The total number of titles carried by the company represents roughly 90\% of platform suppliers' own claims about the total number of applications available for their platforms. This suggests the data offer very good coverage of the total supply of applications to the market.

The data presented in this study reflect the availability of products and transactions conducted by the firm from 4Q 1999 to end of year 2004, at monthly increments. Data include monthly lists of available titles, the firms that developed them, the platform for which they were designed, an indication of the category of software, and basic sales data. The data set includes 9 platform categories (including a "not identified" category where the title has not been identified as working with a particular platform). These titles are divided into 12 categories of software types (again, including a "not identified" category). To isolate the sales of software applications to individual consumers, I remove a number of observations from the data set. I also delete titles that refer to digital content (songs, graphics, ebooks, etc.) that are not applications. The data also includes some several hardware devices and so these are removed. I also eliminate server-side software and large enterprise contracts. This leaves 140,892 title-quarter observations, corresponding to 14,839 unique titles.

I supplement these data with other estimates of industry-level measures. These were assembled by combining secondary sources and information provided by industry participants to produce estimates. Descriptions of these data and sources are described in the main discussion to follow. Ample descriptive statistics are provided in the descriptive analysis to follow.

4.5 Descriptive Analysis

4.5.1 An Industry Structure Conducive to Network Effects

Patterns in the data are consistent with the expected effects of growing entry to software developer networks and many of the assumptions underlying these points, as described in Section 2.

\textsuperscript{13}It is difficult to get precise market share data, but with roughly 2.6 million transactions in 2004, and an installed base of mobile computer users of roughly 25 million (outside Japan), this is one transaction for every 10\% of consumers. Market researchers have anecdotally suggested that roughly 20\% of all users have downloaded third-party software.
Nonetheless, the data also suggest ample scope for the importance of individual firms’ strategic investment incentives to affect the supply of software to end-users.

To begin, price declines over time coincide with the growing size of developer networks. Whatever link may exist between growing entry and price declines, the change in price that might be caused by intensifying competition appears modest. Average nominal prices of applications software decline by around 15% per year (see following figure). However, price adjustments, removing the “fixed effect” of each title to were far smaller and sometimes even positive. Given relatively flat or increasing trends in price adjustments, any effect of entry on price is likely limited in magnitude. Even without knowing demand elasticities in this market, it may be reasonable to conjecture that the relatively small fraction of overall cost of ownership of these systems these one and two dollar annual changes represent, suggest that price changes were not a central factor in stimulating demand and reinforcing the network effect.

---

14 Price adjustments were calculated by removing the initial price of a product upon entry.
15 The price declines on the order of $10 over 5 years are likely dwarfed by the costs of possibly two hardware devices over this period of time, perhaps $500-$1000.
16 Nair et al (2004) did not consider the effect of software price decline in their empirical analysis of the indirect network effect in this industry.
*Includes only the effects of price adjustments and not the effect of new product releases*

Figure 4-4: Trends in the Size of Developer Network and Price of Software Applications
A much more dramatic trend in the supply of software was the rapidly growing library of software titles that came available as networks of software developers grew. This expanding selection resulted in a combination of growing availability, variety and quality. While there are clearly new versions of software launched, reflecting quality advances, and price declines associated the launch of new products (discussed above) suggest possible cost declines the growing selection most obviously reflects expanding variety of titles. This is consistent with the thrust of theory and empirical research on indirect network effects, and earlier studies of this industry in particular (McGahan et al. 1997; Nair et al. 2005). The following figure begins to suggest this variety, revealing the hundreds and thousands of titles that appear within any of several categories. (This breakdown was relatively stable from year to year.) Visual inspection of the titles reveals a great many subcategories and distinct applications types within each of these categories, along with numerous titles obviously competing head-to-head. Within the top 50 titles in each platform, which were inspected in detail, appeared a mixture of titles reflecting a significant proportion of titles from each of these subcategories (with some exceptions, such as Java, which had fewer business-oriented applications and more simple games and sounds.) In the categories somewhat distinct subcategories of applications could be identified—such as calculators, calendars, clocks and timers, spreadsheets, word processors, chess games, web browsers, etc.—amongst a larger continuum of less easily grouped titles. This relatively continuous distribution of titles, with sometimes direct and indirect competition among titles seems to reflect the conceptualization of the differentiated product space described by the theory (Section 2).

This generation of an extraordinary level of variety appears to have been valued by end-users. The distribution of demand across the many titles supplied for each platform that end-users had a "taste for variety." As an indication of "long tail" distribution of demand across titles, the quantity sold for individual titles in the Palm, Microsoft, RIM, Symbian, Linux and Java platforms is plotted below according to rank by sales. To provide a meaningful measure of the relative size of sales in relation to other titles available for the platform at a given point in time, the statistic plotted is the share of quantity sold of all titles in a platform, averaged over each of the periods it was for sale.17 In grouping together data for lower-ranked titles, these titles

17 Because of this averaging, the actual values may not sum to 100%.
Figure 4-5: Wide Variety of Complements Supplied

were first individually ranked in terms of their average share of sales. Once ranked, their sales were then added together to calculate their total share of quantity sold. and then averaged over each of the periods in the sample. These statistics are presented for each firm and group of firm, ranked by average market share.

As one might expect, the leading titles far outsell follower titles when looking many of the platforms. The shares of leading titles in Microsoft, Java, Palm and Symbian averaged between 3% and 22% of all units sales. Sales for individual titles after the 100 most popular titles drop to near negligible sales per title. However, in the larger developer networks of Palm and Microsoft, the market share of leaders is far from dominant. Further, as a group, the lower-ranked titles in fact contribute a sizable fraction of total software sales— even those titles ranked 1000 and above (for those platforms for which there are more than 1000 titles). Therefore, the distribution of demand across the vast supply of variety is very much consistent with the usual conceptualization of network effects.

Also consistent with the view of the importance of entry of large numbers of developers to produce this variety is the relatively low concentration or tendency to concentration in the industry. The following figure plots the market share of top developers in the Palm system,
ranked from first to one-hundredth ranked firm. The cumulative market shares for different years are shown, as well as the average market shares for each rank of firm on average over the sample period. A first important observation is that the average market shares of the market leaders are quite small—on the order of 5%. A second observation is that the trend is towards less concentration rather than more concentration over time. Therefore, there do not appear to be compelling returns to scale or other sources of positive feedback in market share, consistent with the conceptualization of the production of applications software in indirect network effects.

4.5.2 A Large Scope for Discretionary Investment Decisions in Determining Overall Software Selection

Notwithstanding a market structure that appears amenable to network effects, features of the industry also suggest ample scope for investment incentives to play a role. A close inspection of the data suggests that the majority of title releases are not from new entry, but from incumbent developers who make decisions to both extend and upgrade their product portfolios. A share of 89% of new titles come from firms who release multiple titles. These developers releasing multiple titles released an average of 7.13 titles within the sample period. The remaining roughly half of the developers released a single title. It may remain the case that these small fringe
Figure 4-7: Relatively Symmetric, Small Suppliers of Software Applications with Little Tendency to Concentration
firms hold the potential of having hit titles. Further, 41\% of titles released by these multi-releasors were estimated to be upgrades and extensions—contributing to the quality of software.\textsuperscript{18} Subtracting the first time releases of these multi-releasor firms, this suggests that roughly half of the titles released by these firms (themselves, roughly half of developer firms) were extensions or upgrades, while the remainder contributed to growing variety of titles. Further, leading firms not only release more titles, but also sells more volume (in the largest developers networks—those of Palm and Microsoft—the top 100 firms of many thousands of developers serve roughly 3/4 of all demand) while charging higher prices—suggesting they might also have higher quality titles.

![Proportion of Unique Title Releases](image)

Figure 4-8: The Decision to Release Multiple Titles has a Greater Impact on the Number of Titles than Does New Entry

It appears that even within these databases of commercially sold software\textsuperscript{19} there is a fringe of firms with few sales and contributing titles of lower quality. Although this group apparently contributes far less to the quality and variety of titles made available to end-users and as individual firms capture negligible demand, as a collective they manage to capture a significant share of demand (roughly a tenth of units sold). Another potential source of value of this group

\textsuperscript{18}Given there were generally few platform re-releases in any case during the time period and given most platform upgrades maintained backward compatibility, these upgrades are likely to in large part simply reflect the product cycle of the application, itself.

\textsuperscript{19}The population of free software developers might potentially add significantly to this group. Data were not available on this.
is that this population might be a source of potential future "hit" titles, where some fraction of ideas generated by this group reveal themselves to be highly successful (whereupon the firm is no longer part of the fringe). The data suggest there is something of this effect at work, but it is not of an important magnitude; of the thousands of developers in the two largest developer networks, Palm and Microsoft, the vast majority that make their way to the top 25 by sales are well within the top 10% of the developer network. The following figure reflects this by plotting the lowest rank held by any firm in these systems that eventually held a position in the top 25 of their systems. Therefore, it indeed appears that while small fringe firms contribute a non-trivial fraction of software demanded by end-users, this fraction is dwarfed by larger developers, building more software of higher quality.
4.5.3 Do Larger Developer Networks Produce a Better Selection of Software?

If marginal entrants are more likely to be smaller, less prolific and less successful developers and perhaps producing titles of lower quality, it becomes an obvious question to ask whether it is in fact productive to allow these developers in the network. On the one hand, new entrants—however small—may contribute incremental diversity to the supply of complements which may be demanded by end-users or embodied in competing products through spill-overs (even if it has not been the case that there is any significant probability these fringe firms will develop into leading developers). Or, failed and unpopular titles may simply have little impact on the market at all. On the other hand, given there is not a sharp distinction between “core” and “fringe” firms, but rather a long decline of demand across firms, it may be that the thousands of tiny firms contributing applications might—as a group—take some fraction of demand away from higher quality, more expensive applications, while also creating additional downward price pressures on those titles. This could potentially reduce the investment made by incumbent...
developers into these applications or even the likelihood they are created at all. While the descriptive data unequivocally show that large developer networks generate large selections of titles, it is not clear they generate the greatest possible variety and quality of titles. Might a slightly smaller network create a better balance between the benefits of marginal entrants and their possibly dampening effect on incumbent software development?

Ideally, to examine this question, detailed information on quality and variety could be distinguished and perhaps even developer investment levels. However, a weaker and more conservative test is whether larger developer networks really produce more software titles. If greater entry stifles investment incentives, the rate of new title development (reflecting perhaps either quality or variety), should fall with developer network size. This assumes that the rate of appearance of new titles (controlling for the development conditions of the particular time period), should reflect the intensity of investment in development activities.

The following table displays correlations between the rate of new title releases in the systems per quarter and the number of developers in each platform. By counting numbers of titles released, this provides a blanket indication of the generation of variety, availability and advancing quality. Although count of numbers of suppliers will not fully reflect the heterogeneity that may exist in these developer networks, they do reflect the primary measure of the developer networks that platform suppliers tracked, as well as reflect the key concepts discussed in Sec-
tion 2—that entry is should bring benefits by intensifying competition and bringing comparative advantages.

Because we are interested in examining the effect of a growing developer network, but controlling for any exogenous shocks in demand or differences across platforms, fixed effects are included for each year and for each platform. To account for the lag between the initiation of a development project and its completion (when we observe the software appear in the data set), I lag the measure of number of developers in a developer network by a year. Despite the integer nature of the dependent variable, a simple ordinary least squares specification is used given the typical large magnitudes of the dependent variable and also to maintain the simplest analysis of the correlations.

To the extent that an expanding market comes from exogenous and unobserved shocks in market opportunity or productivity of investments\(^{20}\) rather than purely from the network effect arising from additional entry of software developers (as it surely will), the unobserved effect of market expansion should simultaneously increase entry and investment in product development. Therefore the following correlation will likely overestimate the effect of new developer entry on the rate of new title development. Therefore, any finding of a negative relationship between the productivity of software development and the size of the developer network can be interpreted as more meaningful than finding a positive relationship. The following regressions therefore relatively weak test for any negative effects of a growing developer network.

Given the clear correspondence between large developer networks and wide selection of titles (and the widespread belief of this among industry participants and scholars alike of the virtues of a large developer network), it not surprising that the first order correlation between the size of the developer network and the generation of new titles is positive and large—in regressions pooling all data for all platforms (Model 1), using the variation between platforms (Model 2) and models using fixed effects to analyze just the variation within a particular developer network (Model 3). (Given the considerable market and technology variation between different platforms, platform fixed effects are used in all following regressions.).

Adding a quadratic term to the correlation to capture any concavity or convexity in the

\(^{20}\)These might arise for investments by the platform supplier in development tools, for example.
### New Software Titles Developed in a Platform-Quarter (N=135)

<table>
<thead>
<tr>
<th>Specification</th>
<th>Pooled</th>
<th>BE</th>
<th>FE</th>
<th>Quadratic</th>
<th>Nonparm</th>
<th>OLS In-lin</th>
<th>OLS In-lin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model:</td>
<td>(1)</td>
<td>(2)</td>
<td>(3)</td>
<td>(4)</td>
<td>(5)</td>
<td>(6)</td>
<td>(7)</td>
</tr>
<tr>
<td>No. Developers¹</td>
<td>.9**</td>
<td>1.2*</td>
<td>.1</td>
<td>2.13**</td>
<td>.001</td>
<td>-.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(.1)</td>
<td>(.3)</td>
<td>(.1)</td>
<td>(.3)</td>
<td>(.001)</td>
<td>(.2)</td>
<td></td>
</tr>
<tr>
<td>No. Developers²²</td>
<td>-.001**</td>
<td>-7E-7**</td>
<td>0.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(.0002)</td>
<td>(.0000002)</td>
<td>(.02)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Explanatory variable divided into increments²

| 2nd     | 245* |
| 3rd     | 412  |
| 4th     | 915* |
| 5th     | -764 |
| 6th     | -61  |
| 7th     | -8   |
| 8th     | -104 |
| 9th     | -336*|
| 10th    | -150.3510 |

### Other Control Variables

|--------------|-------|-------|-------|-------|-------|-------|-------|

| R-squared | 0.51 | 0.9 | 0.33 | 0.88 | 0.91 | 0.91 | 0.91 |

| Number of platforms | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

Huber-White robust standard errors reported; not reported for incremental dummies

* significant at 5%; ** significant at 1%

(1) All developer variables lagged one year
(2) Dummy variables reflecting the range of No. of developers, divided into 10 increments of equal magnitude from lowest to highest, where a dummy reflects being within or above the increment.

Figure 4-12: Rate of New Title Releases Regressed on Size of Developer Network
correlation finds a strong negative second order relationship (Model 4). The following diagram shows the partial relationship between new title development and the size of the developer network revealing that at the highest levels of the developer network, the generation of new title did not just flatten but it turned negative. The maximum based on the results of the quadratic model is estimated to be at 1062 developers—far short of the domain that extent to 2000. While the interpretation of this precise level remains clouded by the inherent upward bias in the correlation, this peak is well within the sample domain. (This scale of developer network includes just Palm and Microsoft, whose data points appear on both sides of the peak.)

A simple non-parametric specification (determining the incremental effect of being within or above the increment) suggests an even stronger negative slope in the data. The coefficient for the 9th interval is statistically significant in indicating a decline in the number of titles developed (Model 5). However, out of the 10 intervals, in fact, the each of the highest 6 intervals take negative coefficients (which are jointly significant), implying a cumulative negative effect of proceeding higher above these levels.21 The following figure plots the residuals in the quadratic specification and the model itself (Model 5), along with the nonparametric model (Model 6)

21 This effect is adequately strong that a regression of the absolute number of titles (rather than rate of releases) also finds a significant negative drop (at the tenth interval), not reported.
Partial Relationship between Rate of New Title Development and Size of Developer Network

It remains conceivable that the decline in rate of new title development is the result of declines in the rate of newly entering firms (from a diminishing profit opportunity) rather than the result of fewer product releases by incumbent developers (who might also be responding to a diminished profit opportunity). To more directly assess whether additional entry leads incumbent firms to develop fewer titles, the regression is re-run using just counts of new software titles by only those firms that release multiple titles in the same. This isolates those firms that clearly reveal themselves to be of a type to develop multiple titles. This selection of firms based on repeated releases, however, effectively biases the sample by effectively selecting on the dependent variable. In attempting to focus the analysis, this selection will likely introduce further positive bias into the measured correlations. For example, if a “multi-releasor” type of developer were to appear in the sample, but was thrown out because the competition brought on by additional entry led this developer not to release a second title, then this negative effect would not be captured using this sampling strategy.

Even with this additional source of positive bias, the substantive result of a negative re-
The relationship between new title development and size of developer network size repeats itself in these regressions. Both quadratic specification (Model 1) and nonparametric (Model 3) specifications find a negative slope as the network size grows large. A test of whether the growing multi-homing of developers on multiple platforms or the growing selection of Java software working on platforms might create possible biases in these results, variables reflecting these trends were included in the quadratic regression (Model 2) and were found to be statistically insignificant.

| New Software Titles Developed by Multiple-Release Developers in a Platform-Quarter (N=135) |
|---------------------------------------------|---|---|---|
| Model: Quadratic Quadratic Nonparm         |
| Specification                               | (1) | (2) | (3) |
| Size of Complementor Network                |     |     |     |
| No. Developers                              | .6** | .4** |     |
|                                              | (.1) | (.1) |     |
| No. Developers^2                            | -.0003* | -.0002** |     |
|                                              | (.00003) | (.00004) |     |
| Explanatory variable divided into increments\(^2\) |     |     |     |
| 2nd                                         | 56*  |     |     |
| 3rd                                         | 153* |     |     |
| 4th                                         | 47   |     |     |
| 5th                                         | -40  |     |     |
| 6th                                         | 50   |     |     |
| 7th                                         | -18  |     |     |
| 8th                                         | -54  |     |     |
| 9th                                         | -11  |     |     |
| 10th                                        | -126 |     |     |
| Other Control Variables                     |     |     |     |
| % Multihoming Developers                    | 69   |     |     |
|                                              | (128) |     |     |
| No. Java Applications                       | .05  |     |     |
|                                              | (.05) |     |     |
| Year Dummies                                |     |     |     |
| Sign.                                       |     |     |     |
| Platform dummies                            |     |     |     |
| Sign.                                       |     |     |     |
| R-squared                                   | 0.96 | 0.97 | 0.92 |
| Number of platforms                         | 8    | 8    | 8    |

\(Huber-White\) robust standard errors reported; not reported for incremental dummies

* significant at 5%; ** significant at 1%

(1) All developer variables lagged one year

(2) Dummy variables reflecting the range of No. of developers, divided into 10 increments of equal magnitude from lowest to highest, dummy reflects being within or above increment.

Rate of New Title Releases Regressed on Size of Developer Network, Multi-Releasor Firms Only
4.6 Discussion and Interpretation: The Role of Incentives in a Network of Complementors

This preliminary investigation begins to suggest patterns consistent with the important role of maintaining investment incentives of component suppliers—even in a “business ecosystem” in which network effects are at work. In this case, the widening selection of software titles for a platform comes chiefly from the discretionary investments of individual incumbent developers rather than incremental entry. Moreover, the negative correlation between the rate of new title development and a large network size (despite an inherent positive bias in this regression) suggests that growing entry of complementors may lead to a crowding-out of investment incentives in new product development, overwhelming the market expansion effects of additional complementor entry. This is consistent with research on multi-sided markets and indirect network effects which argues that despite positive externalities acting between complementors and end-users, that interactions among complementors are competitive and the net effect of entry can
be to reduce the profits of complementors (Church and Gandal 1992; Economides 1996; Ellison and Fudenberg 2003; Rysman 2003; Hagiu 2004; Economides and Skrzypacz 2004, Augereau et al. 2006).

This combination of issues—a market expansion effect, competitive crowding, differentiated entry, and investments in product development—can be illustrated by simply modifying the standard model of differentiated entry suggested by Katz and Shapiro (1994) to understand the effects of entry in component markets. (See Section 2.) Modifications include allowing for expansion of the market opportunity with greater complementor entry and incorporating investments in product development—all within a standard model of differentiated entry on a circle (Salop 1977). This depiction is intended solely as an exemplifying model to fix ideas about how this combination of issues could lead to the observed patterns in the data, rather than a general treatment of the issues.

Consider a differentiated product space represented as a circle of circumference $S$ around which $n$ firms locate equidistantly. The circle has demand of density one, and there are “transportation costs” $t$ of consumers travelling a distance, $x$, to purchase from a given supplier. (High transportation costs are also customarily interpreted as the extent of differentiation between entrants, with transportation costs of zero reflecting head-to-head competition without differentiation.) Apart from their horizontal differentiation, the product offering of entrants conveys a benefit, $U$, to end-users. In relation to the empirical context, we may consider increases in $U$ to reflect product. Because this formulation considers entry to be entry of firms rather than entry of products, $U$, may also be interpreted informally as localized product proliferation around the point of entry, although this is an indirect interpretation. I consider the size of the market opportunity to expand with entry of complementors, $S = S(n)$ where $S'(n) > 0$. This is the standard assumption of the literature. (This is a conservative assumption in relation to the point illustrated here that the value of the complementor network can decrease with growing entry.) Demand for firm $i$ is given by the standard result:

$$D_i(p_i, p_{-i}) = 2x = \frac{p_{-i} - p_i + U_i - U_{-i}}{t} + \frac{S(n)}{n}$$

Each complementor entrant can make investments, $z$, in product development, leading to an increase in $U$. Thus firm profits are given by: $\pi_i = (p_i - c_i)D_i - z_i$. I consider the investment has
a diminishing impact on product quality and takes the following form: \( U_i = z_i^{-\gamma}, 0 < \gamma < 1. \)\(^{22}\) Solving first order conditions in price and in product investment, assuming equilibrium price and investment are then equal to those of other firms in a symmetric equilibrium and solving equations simultaneously, the expression for investment by individual complementors is given as follows:

\[
z^* = \left( \frac{\gamma t S(n)}{n} \right)^{\frac{1}{1-\gamma}} = \left( \frac{\gamma t}{n/S(n)} \right)^{\frac{1}{1-\gamma}}
\] (4.1)

This exemplifying model shows a response of individual investment to growing entry to that depend positively on the productivity of investments, level of differentiation, and the market opportunity, as would be expected. The effect of increasing entry—perhaps through reducing fixed costs of entry, through subsidies or widening of licensing—will depend on the relative strength of the (positive) market expansion effect and (negative) competition effect. A sufficient statistic for the effect of entry in this exemplifying model is the ratio of the number of firms to the market opportunity, \( n/S(n) \), or the "crowding" in the complementor network. Where the network effect is not so strong as to stave-off crowding, individual investments will decline in this formulation. Total amount of investment, \( Z = n \times z \), will can also decline in this model, where the contribution of investment by the marginal entrant does not outweigh the crowding effect it has on each of the incumbent complementors. Clearly additional investigation of the role of investment incentives in multi-sided markets is warranted. This illustration is simply intended to clarify in a straightforward and familiar context the possible stifling effects of entry on product development—even in the presence of indirect network effects.

4.7 Conclusion

The literature on indirect network effects and multi-sided markets suggests that a platform supplier will do well to promote the creation of complements to its platform. More complements and product selection begets more demand; a larger installed base of buyers begets more complements. The result should be more value created around a platform and a greater oppor-

\(^{22}\)This form of investment function is used by Vives (2005) in illustrating the impact of growing competition on innovation incentives in an entire industry, rather than within the subeconomy around a particular platform.
tunity for the platform supplier to appropriate a fraction of that value. Thus, it has become widely understood that platform suppliers should attempt to grow large “business ecosystems” around their platforms so as to foment such virtuous circles of end-user adoption and the supply of complements. Famous examples of such successful business ecosystems provide ample proof that such a strategy indeed has its virtue. Following this logic, it is now orthodoxy for suppliers of computer platforms to aggressively promote huge networks of application developer around their platforms, through a number of means. However the idea that facilitating, even promoting the entry of complementors should be the mechanism through which a greater variety and better complements should arrive has never been formally developed in theory, nor has it been systematically tested empirically.

For the most part, theoretical work focuses on studying the implications of indirect networks a growing supply of complements and growing installed base of users, without deeply studying mechanisms. Empirical research on indirect network effects has shown examples where greater variety of complements promotes demand, but has not studied whether more complements requires more complementors. It might be, however, that a single large complementor is better because of say, extreme economies of scale, or superior “capabilities” and comparative advantages at producing the complement, but still a growing installed base creates incentives to release more complements. It may be that greater numbers of complements create more market confusion. Or perhaps less extraordinarily, more entry may simply produce competitive crowding and reduce investment incentives of complementors (who tend to compete with one-another, despite the indirect network effects that may operate between themselves and end-users). Notwithstanding enormous recent progress in empirical research on indirect network effects, we still do not know very well how the supply-side of these network effects work. Consequently, we can not yet understand the best way to organize and manage networks of complementors beyond simply encouraging their growth.

This paper amassed detailed firm and product-level data on the generation of new software titles in leading mobile computer platforms. This industry conforms to the typical model of aggressively promoting the entry of software developers around a platform. Because the data were at the level of individual titles and releases and listed developer firms, as well as measure of demand for these titles, they allowed an especially detailed review of both the structure of
demand and supply across these titles. I found that, as theory predicts, and econometric studies of aggregate levels of demand suggest, that indeed demand is distributed across a great many titles, consistent with a “taste for variety” that might motivate stimulating such a great many titles. However, on the supply side, I find that rather than a direct correspondence between the entry of developers and the creation of new titles, I in fact find the vast majority of titles are produced by incumbent developers who have chosen to release multiple titles in succession. Further, it appears these firms with multiple releases not only contribute new titles, but also upgrades and extensions to existing titles. These revelations throw light on the importance of the discretion taken by individual complementors in producing a succession of titles. The production of multiple titles by a subset of firms has a greater impact than the contribution of entry as such, given that a vast majority of titles are not the first releases of the developers. This importance of the discretion of complementors in making investments in product development raises the question of whether additional expansion of the complementor network may in fact have a negative impact on the production of new titles, given the additional competition this entry may generate. Regressions of the rate of new title releases in networks and the size of software developer networks, controlling for the platform and for the time period, find a negative correspondence between the rate of new title releases and a large complementor network. These results are interpreted as the exhibiting the importance of maintaining the investment incentives of complementors.
Bibliography


171


