The role of product complexity and firm competency in the diffusion of user-customized systems[*]

Benedikt Gamharter, Tobias Kretschmer

Institute for Communication Economics (ICE)
Munich School of Management, University of Munich
Schackstr. 4/III
D – 80539 Munich
benediktg@gmx.net
t.kretschmer@lmu.de

Abstract: We study firm-level adoption of packaged software products across 3,891 sites in the United Kingdom over four years (2000-2003). We treat the entire bundle of software in a firm as its software product and divide the bundle into related, but distinct subsystems. Diffusion is studied across those subsystems. We introduce three factors that may affect the adoption decision: product complexity, architectural linkage and competency destroying/enhancing scale. We find that firms enhancing their competency are more likely to adopt new technologies, while there is a negative relationship between architectural linkage and product adoption. We also find that changes in the complexity of core subsystems hinder switching.

1 Introduction

This paper studies the adoption of software in the United Kingdom (UK) from 2000-2003. We extend the literature by developing an adoption framework based on firm-technology characteristics to study the adoption of user-customised technology. Our central research questions are: a) Which firm characteristics affect the adoption of specific software? b) How is the adoption decision of a particular product affected by the concurrent use of other, complementary products? c) Do firms upgrade their software system 'in bursts', i.e. many changes at once, or gradually, i.e. continuous, but small changes? Our key findings are that firm competencies and the complexity of the current software bundle play an important role in firm adoption behaviour. Specifically, firms with higher levels of competency are more likely to switch technologies, while switching is less likely in firms with highly complex software systems already in place. Our study

[*] This work is part of the project "Explaining Productivity and Growth in Europe, America and Asia" funded by the Anglo-German Foundation, whose support is gratefully acknowledged. We would also like to thank participants from the "*The knowledge-based society* workshop" (sponsored by the Research Council of Norway 2006-2009) for their comments on an earlier version of this paper.

is the first to explicitly recognize the fact that complex products are mostly user-customized, which has implications for the adoption of new system components.

2 Existing literature

The literature on the diffusion of new goods and services can broadly be classified in single-technology and system-technology studies.[1] While the s-shaped diffusion curve is universally accepted, there is no clear consensus on what generates it.[2]

Many diffusion studies consider single, mostly manufactured products or services.[3] The focus of these studies is on identifying adopter or firm characteristics that have an effect on the likelihood and/or timing of adoption and therefore the speed of diffusion. More recent studies include the effect of installed base on diffusion speed, especially for technologies with network effects. Lack of a sufficient installed base can delay adoption and therefore diffusion [FS86], although once critical mass is reached, diffusion can be rapid [Ca90]. Studies looking at the diffusion of single technologies are appropriate in contexts where adoption also takes place in isolation. Similarly, with direct network effects the need to consider related technologies is limited. However, new technologies do not always appear in isolation, but as systems of complementary products.

A system of complementary products frequently (or exclusively) used jointly can have important implications for diffusion. In the simplest case where complementary products are used in a constant ratio, treating the system as a single technology will not bias results. However, if the ratio is variable, or if diffusion of one technology depends on the availability of the other, measuring diffusion of both components is necessary to study the true mechanisms behind the system's diffusion. This is especially relevant for technologies with indirect network effects. Gandal et al. [GKR00], Nair et al. [NCD04] and Gupta et al. [GJS99] are examples of studies where cross-(or indirect) network effects are estimated. The key questions in that research are the strength and direction of indirect network effects between complementary goods. However, the products studied are still relatively simple since the technology and the nature of complementarities is determined by the producer. That is, adopters still purchase a well-defined product (e.g. CD players requiring CDs). For many technologies, however, there is a significant degree of user-driven customization of the bundle of products. The diffusion literature has not yet addressed this aspect of complex technologies. We now introduce a framework to help organize these effects of user-customised technology on diffusion.

[1] We use the term 'technology' as a catch-all for all goods and services.
[2] Geroski [Ge00], Stoneman [St02], Nelson et al. [NPS04] and Rogers [Ro95] review the different models of technology diffusion. Karshenas and Stoneman [KS93], Zettelmeyer and Stoneman [ZS93] and Grajek and Kretschmer [GK06] provide comparative analyses of the different models.
[3] The range of products studied is enormous and we will not attempt a review of the existing studies. For a partial list of studies, see Geroski [Ge00], Stoneman [St02], or Rogers [Ro95].

2.2 Diffusion of user-customised technology

The IT infrastructure used by a graphic design firm will be very different from an international bank's or a biotech company's. Although most companies use IT, there are significant differences between users which may affect their adoption behaviour. However, it is the user who defines this technology or system. There is a complex relationship governing the diffusion of user-customized systems, but there are also significant adjustment costs arising from introducing a new component. To our knowledge, no studies explicitly look at the diffusion of such user-customized systems.

We borrow from the literature on mix-and-match products [MR88, MR92] and modular systems [Sc00, HC90]. This literature claims that the usefulness of a product depends on the other technologies currently used in the firm. Kretschmer [Kr05] terms this internal complement effects (ICE). The concept of ICE stresses the notion that it is not the global availability of complementary products (as typically assumed in the literature on indirect network effects), but the actual usage of technologies in the firm that affects adoption of a modular good. To develop an adoption framework for such goods, we turn to the literature on the innovation of modular products and complex technologies.

The innovation literature has long recognized that new products are flexible in their specifications of components and their linkages.[4] As an IT product is developed, individual components are changed and communication interfaces are defined. Henderson and Clark [HC90] and Schilling's [Sc00] work on innovation in modular systems explicitly studies the system architecture. While it is intuitively appealing that subsystems are linked and that the nature of these linkages will determine the end product, operationalization of these concepts has often proved difficult, both theoretically and empirically. We therefore use Gatignon et al's [GTSA02] approach to describe and analyze complex innovations and adapt it to the study of user-customized system adoption. Gatignon et al.'s approach to evaluate innovation states that:

> […] innovation can be comprehensively described by distinguishing between product complexity (the number of its subsystems), the locus of the innovation in a product's hierarchy (core/peripheral), different types of innovation (generational and architectural), and the innovation's characteristics (incremental/radical, competence-enhancing, and competence-destroying) [GTSA02, pp. 1104].

We follow their approach and extend it by including product complexity as a fourth innovation characteristic (see Table 1). In their study, however, the framework is used to predict the origins of new product innovations, i.e. *supply* of a new product, rather than *demand* for the product and its components. We ask how likely a firm with a given system architecture (described in Table 2-1) is to adopt a new technology.

---

[4] For a detailed review of the earlier literature on innovation see for instance Utterback [Ut94] and more recently Christensen and Raynor [CR03], Markides and Geroski [MG05] and Moore [Mo05].

Characteristic		Definition
Locus of innovation		‘Core subsystems are those that are tightly coupled to other subsystems. In contrast, peripheral subsystems are weakly coupled to other subsystems [GTSA02, p. 1106].’
Product complexity		The product complexity increases in the number of subsystems the product is made up of.
Innovation type	Generational consolidation/expansion	‘Generational innovation involves changes in subsystems linked together with existing linking mechanisms [GTSA02, p. 1106].’
	Architectural linkage	‘Architectural innovation involves changes in linkages between existing subsystems [GTSA02, p. 1106].’
Innovation characteristics	Competency destroying/competency enhancing	‘Competence-enhancing innovation builds upon and reinforces existing competencies, skills, and know-how. Competence-destroying innovation obsolesces and overturns existing competencies, skills, and know-how [GTSA02, p. 1107].’
	Radicalness	‘Incremental innovations are those that improve price/performance advance at a rate consistent with the existing technical trajectory. Radical innovations advance the price/performance frontier by much more than the existing rate of progress [GTSA02, p. 1107].’

Table 1: Structural approach to evaluate innovation

Gatignon et. al. [GTSA02] construct scales to assess the characteristics in Table 1. Conversely, we use measures in our data to proxy for these characteristics.

3 Product framework

We follow the definitions from International Data Corporation (IDC) and identify the following product categories: Application software, application development & deployment tools and system infrastructure.[5] We focus on professional software and especially on Operating Systems (OS) within the system infrastructure segment and on Desktop Applications (DA) and Enterprise Applications (EA) in applications software.

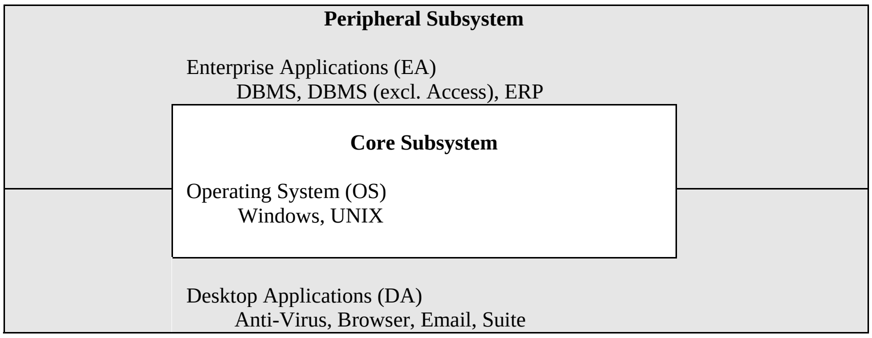

Figure 1: Software product, subsystems and hierarchy

[5] IDC is one of the largest market intelligence firms covering the world-wide high-tech market. See their website for more details: http://www.idc.com

A firm will purchase software to improve productivity and meet its business objectives [BG99, BK96]. We treat a firm as being composed of different tasks or functions [Kr04] and assume that software is chosen to match the respective task requirements. Assuming that there is a central decision maker in the firm, decisions are made at the site level to optimize site performance and *software applications are not viewed in isolation*, but rather in what we term a firm's *software architecture*, which is made up of subsystems, i.e. individual software. Hence, a firm faces a user-customized adoption decision.

Figure 1 classifies a firm's software product and lists the individual products in our data. At the heart of the product are core subsystems characterized by multiple connections with other subsystems and their importance to the overall system. Further, the software product consists of peripheral subsystems. We define two types of peripheral subsystems: *DA* are generic, typically not customized and used across the firm. *EA* are also peripheral, but more complex subsystems within the product hierarchy as they are mostly customized and affect numerous related systems and processes.

4 Data and conceptual framework

4.1 Data

The data we use is from Harte-Hanks Inc. (HH). HH is an international direct and targeted marketing company which collects annual information on companies' IT stock.[6] We use the HH UK dataset from 2000 to 2003. HH collects information on roughly 16,000 sites. We eliminated observations with data inconsistencies.[7] This cleaning process left 23,639 observations. We then balanced the panel with a total of 15,564 observations from 3,891 firms. Of the firms we study, 49% are in manufacturing, 42% are service firms and 9% are other sites. 87% of the sites are located in England, 7% in Scotland, 3% in Wales and the rest in Northern Ireland or other areas (e.g. Channel Islands). 35% are large-sized sites (200-499) and 34% are medium sized firms (50-199). 25% of sites are very large (500+ employees) and 6% are small (1-49 employees).[8]

4.2 Software and site characteristics

We construct the following product hierarchy: three generic product categories (**PC_***) made up by product families (**PF_***). Product families are defined as groups of close substitutes. Product families consist of individual products (**PD_***) (see Table 2).[9]

---

[6] We do not distinguish between firms and government institutions. The terms 'firm' and 'site' are used interchangeably and address the entire sample.
[7] This paper follows the data cleaning and panel data construction process as set out by Kretschmer [Kr05].
[8] Please contact the authors for detailed descriptive statistics.
[9] The variables names follow a consistent logic. For instance, 'PF_ERP' refers to 'Product Family ERP'. When we want to refer to all variables of a category we can do so; for instance, 'PF_*' refers to all Product Families or 'IND_*' to all Industry Groups. Please contact the authors for a detailed description of how the variables are constructed.

PC	Operating System (OS)	Desktop Applications (DA)	Enterprise Applications (EA)
PF	- Windows - UNIX	- Browser - AntiVirus - Email - Suite	- DBMS - DBMS (excl. MS Access) - ERP
PD	- 9 Operating Systems (Windows XP, etc.)	- 2 Browser; 5 Anti-Virus; 4 Email; 7 Suite (Explorer, Outlook, Office 97 etc.)	- 7 DBMS; 5 ERP (Oracle, SQL Server, etc.)

Table 2: Product variables

All product usage variables are dummy variables = 1 if the site uses the software in any given year and = 0 otherwise. This data setup allows us to study the dynamics of product change.[10] We construct variables on site characteristics along the following dimensions:

- **Industry groups**: Sites are classified based on their primary activity in a) Service Industry (**IND_G_Service**), b) Manufacturing Industry (**IND_G_Manuf**) and c) Other (**IND_G_Other**);
- **Firm size**: We include a variable (**LOG_Emp**) describing the size of the site;[11]
- **Technological expertise**: We capture a site's IT expertise via four variables. A firm is more IT intensive the higher the proportion of IT staff (**IT_Intensity**), and the higher the proportion of IT developers (**IT_Dev**).[12] We also capture the number of servers (**LOG_Server**) on site as well as site connectedness (**LOG_Network**). We also take these as proxy for IT expertise.

4.3 Adoption characteristics

In our framework we combine the idea of evaluating innovations and our product framework. We ask if firm characteristics help estimate the probability of future adoption. In addition to the three firm characteristics introduced above, we introduce structural change variables (**SC_***). The variables compare the year-on-year change in firm characteristics. Hence, the t-1 variables are calculated based on the results between *t-2* and *t-1*. Here, we study if past changes in the firm have an impact on further changes. The change variables are calculated as set out in (1).

$$(1) \qquad SC\_*_{\Delta} = \log\left(\sum \frac{SC\_*_{t}}{SC\_*_{t-1}}\right) \text{ (t = time of adoption)}$$

We also introduce product complexity variables (**PC_***) which are level variables. Table 3 summarizes our construction of our **SC_*** and **PC_*** variables. We use a hazard rate specification to study the product adoption decision in Section 5.1.

[10] A detailed description of how the variables are constructed is available from the authors upon request.
[11] As discussed earlier, we do not want to explore/include the complexities of organisational design into our analysis. Hence, for the sake of simplicity, we do not differentiate between potential site types (e.g. headquarters, subsidiaries) and assume that all sites can influence their software adoption decision.
[12] See Kretschmer [Kr04] for an earlier operationalization of these variables.

Characteristic[13]	Measure (Variable)
Locus of innovation	We study the effect of core subsystem replacement or enhancement. The paper distinguishes between Microsoft and UNIX products (**SC_Locus_Core_MS** and **SC_Locus_Core_UNIX**). We also study the effect of a peripheral subsystem being introduced (**SC_Locus_Peri_***) and study the various product families. The variable flags when sites adopt a new technology. Those are the dependent variables for our adoption analysis.
Product complexity	We define an upper and a lower bound of product complexity. **a) Lower bound:** We simply count the subsystems the product is made up of. We can not match application software to OS and hence we do not know if application software, for instance an Anti-Virus product, might need to be multiple-configured in order for it to work across multiple OS. Therefore, in order to capture product complexity conservatively, we derive three complexity variables describing the complexity of core (**PC_Complex_Core**) and peripheral subsystems (**PC_Complex_Peri_DA, PC_Complex_Peri_EA**). **b) Upper bound:** Again, product complexity is the count of the subsystems used. Here, however, we consider that peripheral subsystems may be used in all core subsystems (**PC_Complex_Product**). We assume the most complex setup of the product and assume that application software is configured for all OS used on site.
Complexity change	We construct the change variable as set out above. The variable is calculated for both **SC_Complex_Change_LBase** and **SC_Complex_Change_UBase**.
Innovation type: Architectural linkage	We construct a variable capturing the change of linkage between subsystems (**SC_Architecture**). We capture architectural linkage, change in linkage between subsystems, change in the way subsystems interact, etc., through the number of network connections (**LOG_Network**) and the number of servers (**LOG_Server**) used at the site. We assume that subsystems interact more if with more network connections or more server power.
Innovation characteristics: Competency destroying/ enhancing	Gatignon et. al. [GTSA02] use two concepts: a) 'New competence acquisition' and b) 'Competence-enhancing/destroying'. We can not do this. However, we test if the competency level of the firm changes by calculating **SC_Competence** based on the IT intensity ratio (**IT_Intensity**) and the total number of IT development employees (**IT_Dev**).

Table 3: Adoption framework

5 Results & interpretation

5.1 Adoption regressions and results[14]

We study adoption of software across three lenses: OS, DA and EA.[15] Our dependent variable is a dummy variable and the regressors are both continuous (e.g. **IT_Intensity** or **IT_Dev**) and discrete (e.g. **IND_G_Service** or **IND_G_Manuf**). To estimate the probability of adopting a certain technology we use a hazard-rate specification (e.g. we are concerned with firms changing their product complexity by adding or replacing subsystems). We use a Weibull hazard rate model as defined in (2).

[13] The variables 'Generational consolidation / expansion' and 'Radicalness' are not included in the framework.
[14] We follow Karshenas and Stoneman [KS93] and Kretschmer [Kr05] who use the hazard rate approach to study the adoption/switching behaviour. This study does not differentiate if a site changes its product configuration or is a completely new adopter.
[15] Regression results for individual OS products are available from the authors upon request.

$$(2) \qquad h_i(t) = e^{\sum_{v=2}^{V} \beta_v D_i + \sum_{w=V+1}^{W} \beta_w X_i}$$

Our Weibull regression results are presented below, but other specifications give similar results. In Table 4 to 6 we use lower bound product complexity, but our results are robust to using upper bound product complexity.[16]

We use a balanced panel. As discussed previously, we calculate the innovation characteristics and identify sites adopting a new technology. To avoid endogeneity problems, we do not study the firm at the time of adoption *t* but their characteristics in the previous year (*t-1*; e.g. the **SC_Locus_*** variables are set = 1 at t-1 if firms switch into the technology at time *t*). Results from the hazard-rate specifications are presented in Table 4 to 6.

	PF_Anti-Virus		PF_Browser		PF_Email		PF_Suites	
Variable Name	Coeff.	S.E.	Coeff.	S.E.	Coeff.	S.E.	Coeff.	S.E.
PC_Complex_Core	.966	.027	.982	.038	.943	.041	$.927^{*}$	.030
PC_Complex_Peri_DA	1.097^{*}	.050	1.011	.080	1.087	.083	.960	.047
PC_Complex_Peri_EA	$.945^{\dagger}$	.029	$.891^{*}$	.043	$.850^{**}$	.043	$.917^{**}$	.030
SC_Complex_Lbase	.791	.125	.859	.189	.859	.178	$.666^{**}$	.099
SC_Architecture	$.592^{*}$	.136	$.178^{**}$	.099	$.359^{\dagger}$	.217	.684	.241
SC_Competence	.914	.058	$1.229^{\dagger}$	.148	1.117	.142	1.014	.102
IND_G_Service	.857	.090	.864	.130	.977	.159	1.035	.115
IND_G_Manuf	.922	.099	.869	.136	.889	.148	1.054	.121
LOG_Emp	1.047	.047	1.141^{*}	.070	1.436^{**}	.104	$1.091^{\dagger}$	.054
IT_Intensity	2.561	1.006	3.944^{**}	1.81	.502	.269	1.415	.501
IT_Dev	.822	.477	$.192^{\dagger}$	.170	55.254^{**}	70.005	.628	.451
LOG_Server	$.921^{\dagger}$	.041	.961	.060	.976	.068	.961	.043
LOG_Network	1.148	.059	.939	.064	$.764^{**}$	.063	.987	.052
P	3.195		3.140		3.209		3.209	
WALD χ^2	5,972.59		3,801.02		3,769.81		4,845.63	
Observations	5,426		7,087		7,353		5,330	

Notes:
- Preferred regression, hazard rate specification (Weibull model)
- ** indicates 1% significance, * indicates 5% significance, † indicates significance at the 10% level.

Table 4: Adoption Regressions: Desktop Software

[16] Those regression results are available from the authors upon request.

Variable Name	PF_OS_Windows Coeff.	S.E.	PF_OS_UNIX Coeff.	S.E.
PC_Complex_Core	.960	.026	1.145	.065
PC_Complex_Peri_DA	.999	.036	.908*	.074
PC_Complex_Peri_EA	.914**	.022	.961	.047
SC_Complex_Lbase	1.055	.105	.966	.226
SC_Architecture	.662†	.150	1.273	.490
SC_Competence	1.035	.058	1.053	.167
IND_G_Service	.959	.081	.918	.148
IND_G_Manuf	.993	.086	.796	.142
LOG_Emp	1.038	.040	.996	.086
IT_Intensity	2.359**	.753	.260*	.176
IT_Dev	.350†	.204	6.633	9.954
LOG_Server	.991	.034	1.062	.088
LOG_Network	1.000	.042	1.081	.110
P	3.173		3.466	
WALD χ^2	5,724.20		3,195.38	
Observations	4,989		8,605	

Notes:
- Preferred regressions, hazard rate specification (Weibull model)
- ** indicates 1% significance, * indicates 5% significance, † indicates significance at the 10% level.

Table 5: Adoption Regressions: Operating Systems

Variable Name	PF_DBMS Coeff.	S.E.	PF_DBMS (excl. MS Access) Coeff.	S.E.	PF_ERP Coeff.	S.E.
PC_Complex_Core	.897**	.028	.919*	.036	.941	.048
PC_Complex_Peri_DA	1.100†	.055	1.118†	.072	1.037	.086
PC_Complex_Peri_EA	.836**	.031	.788**	.037	.863*	.052
SC_Complex_Lbase	1.035	.152	1.246	.195	.769	.159
SC_Architecture	.898	.261	.5703	.203	.738	.316
SC_Competence	.996	.079	.976	.105	1.073	.127
IND_G_Service	1.008	.114	1.150	.154	.924	.194
IND_G_Manuf	.885	.101	1.036	.140	.836	.169
LOG_Emp	1.043	.047	1.069	.063	.971	.084
IT_Intensity	1.898†	.726	1.998	1.366	.467	.451
IT_Dev	.760	.584	.421	.462	4.661	13.742
LOG_Server	.945	.045	1.022	.059	.942	.085
LOG_Network	1.028	.053	.977	.066	1.117	.116
P	3.139		3.310		3.294	
WALD χ^2	5,280.27		4,180.55		3,073.18	
Observations	5,705		7,280		8,079	

Notes:
- Preferred regression, hazard rate specification (Weibull model)
- ** indicates 1% significance, * indicates 5% significance, † indicates significance at the 10% level.

Table 6 : Adoption Regressions: Enterprise Applications

5.2 Interpretation

We discuss our results in three groups: a) The effect of site and industry characteristics on switching with a focus on IT competency. b) The effect of increased product complexity on switching. c) The consequences of architectural change on switching.

Firm competency and task variety

Higher IT intensity facilitates switching to new software. This result is robust and significant across our specifications and applies to core and peripheral subsystems (with the exception of ERPs). This suggests that IT intensity, as measured by the IT staff on site or connectivity is indeed a measure of a firm's ability to cope with new software. As the interlinkages between software products are high, the costs for switching to a single product are typically much higher than the simple purchasing and adjustment costs for an isolated application, so that IT-savvy firms have higher absorptive capacity with respect to adopting new software. Further, larger firms are more likely to switch. This confirms the results by Kretschmer [Kr04] that larger firms have higher task variety.

System complexity

We find a negative relationship between the complexity of the core subsystem and software adoption (**SC_Complex_Core**). In other words, once a complex system is up and running, its core component is unlikely to be changed. We also find that changes in the complexity of enterprise applications (**SC_Complex_Peri_EA**) hinder switching. That is, if a firm implements a large scale system such as an ERP, further switching is less likely. This highlights the importance of recognizing the links between different software, in particular large-scale enterprise applications. Conversely, a change in the complexity of desktop applications does not have a significant effect on switching in other product families. This is consistent with the intuition that desktop applications do not trigger a long list of additional changes and adjustments if they change.

Architectural change

If system architecture changes, e.g. the firm invests to increase its network connections (**LOG_Network**) or adds servers (**LOG_Server**) to its IT infrastructure, it may try and avoid further software changes in the near future. We can only speculate that there is a time lag between architectural (hardware) change and software adoption. That is, lumpy investments are preferred to multiple smaller ones demanding continuous adjustments. Indeed, we find that architectural change is negatively related to further changes. This is particularly relevant to the adoption of Windows, where a core system change implies that a new round of adjustment is needed.

6 Concluding remarks

We proposed a framework for the adoption of user-customised systems. The literature is focused on well-defined (manufactured) products. However, this may not be appropriate for more complex technologies. We highlighted that software products are harder to describe than other component systems (particularly classic examples such as CD players, etc.), due to their high degree of modularity and complexity. Another observation is the importance of understanding if the demand or supply side defines the product.[17] Existing work mainly assumes that the product design (or 'assembly' of subsystems) is completed by suppliers (which is true for CD players, for instance). However, as we showed for software products, the adopting firm designs the product, which requires a different approach to study adoption. We believe that our framework can be successfully used to refine the demand side of software adoption.

We measured potential adopters' product complexity and complexity change, architectural linkage and competency destroying/enhancing scale and find a positive relationship between firm size and the adoption decision. We also find that changes in the complexity of core subsystems and enterprise applications hinder switching. We also study the effect of architectural change and find a negative relationship between architectural linkage and product adoption. Finally, we find that competence-enhancing firms are more likely to subsequently adopt new technologies.

The challenge for future work is twofold: a) To improve the adoption framework and address current limitations.[18] b) To better understand the managerial consequences derived from this method. The following questions are especially interesting: i) 'How are core subsystems replaced? Are they always replaced by new core subsystems (radical innovation) or can they also be 'graded down' to become a peripheral subsystem?'; ii) 'How can we better describe what core and what peripheral subsystems are?'; and, iii) 'Can peripheral subsystems 'be positioned' to be 'more core' than others?'

We stress that our empirical results are preliminary. Nevertheless, we identified a set of interesting new characteristics to study user-customized adoption decisions. To conclude, we are encouraged by the results, believe that they provide new insights on diffusion of complex products and hope that they motivate future research.

References

[BG99] Bresnahan, T. and S. Greenstein (1999) Technological Competition and the Structure of the Computer Industry. Journal of Industrial Economics, 47, 1-40.

[BK96] Brynjolfsson, E. and C. Kemerer (1996) Network Externalities in Microcomputer Software: An Econometric Analysis of Spreadsheet Market. Management Science, 42, 1627-1647.

[Ca90] Cabral, L. (1990) On the adoption of innovations with network externalities. Mathematical Social Sciences, 19, 299–308.

[FS86] Farrell, J. and G. Saloner (1986) Installed base and compatibility. American Economic Review, 76, 940–955.

[17] We mean 'define' literally in this context and do not refer to the marketing idea of 'supply push' and 'demand pull'.

[18] e.g. a) Data history: analysis would benefit from a longer panel. b) Variable construction: construct technological expertise variables which do not look at internal headcount data only. c) No financial details: map financial information to our current dataset.

[GKR00] Gandal, N., M. Kende, R. Rob (2000) The Dynamics of Technological Adoption in Hardware/Software Systems: The Case of Compact Disc Players. Rand Journal of Economics, 31, 43-61.
[GTSA02] Gatignon, H. and M.L. Tushman, W. Smith, P. Anderson (2002) A Structural Approach to Assessing Innovation: Construct Development of Innovation Locus, Type and Characteristics. Management Science, 48, 1103-1122.
[Ge00] Geroski, P.A. (2000) Models of technology diffusion. Research Policy, 29, 603-625.
[GT06] Grajek, M. and T. Kretschmer (2006) Usage and Diffusion of Cellular Telephony, 1998-2004. NET Institute Working Paper #06-21.
[GJS99] Gupta, S., Jain, D., and M. Sawhney (1999). Modeling the Evolution of Markets with Indirect Network Externalities: An Application to Digital Television. Marketing Science, 18, 396-416.
[HC90] Henderson, R. and K. Clark (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. Administrative Science Quarterly, 35, 9-30.
[KS93] Karshenas, M. and P. Stoneman (1993). Rank, Stock, Order and Epidemic Effects in the Diffusion of New Process Technology. Rand Journal of Economics, 24, 503-528.
[Kr04] Kretschmer, T. (2004). Upgrading and niche usage of PC operating systems. International Journal of Industrial Organization, 22, 1155-1182.
[Kr05] Kretschmer, T. (2005). Competing technologies in the database management systems market. NET Institute Working Paper #05-17.
[NCD04] Nair, H., Chintagunta, P. and J.-P. Dubé (2004). Empirical Analysis of Indirect Network Effects in the Market for Personal Digital Assistants. Quantitative Marketing and Economics, 2, 23-58.
[NPS04] Nelson, R., A. Peterhansl and B. Sampat (2004). Why and How Innovations Get Adopted: A Tale of Four Models. Industrial and Corporate Change, 13, 679-699.
[MG05] Markides, C. and P. Geroski (2005) Fast Second: How Smart Companies Bypass Radical Innovation to Enter and Dominate New Markets. Jossey-Bass, San Francisco, USA.
[MR88] Matutes, C. and P. Regibeau (1988). “Mix and Match”: Product Compatibility without Network Externalities. Rand Journal of Economics, 19, 221-234.
[MR92] Matutes, C. and P. Regibeau (1992). Compatibility and Bundling of Complementary Goods in a Duopoly. Journal of Industrial Economics, 40, 37-54.
[Mo05] Moore, G. A. (2005) Dealing with Darwin: How Great Companies Innovate at Every Phase of Their Evolution. Penguin, London, UK.
[Ro95] Rogers, E. (1995) Diffusion of Innovations. 4th edn. The FreePress, New York, USA.
[Sc00] Schilling, M. A. (2000) Toward a General Modular Systems Theory and Its Application to Interfirm Product Modularity. Academy of Management Review, 25, 312-334.
[St02] Stoneman, P. (2002) The economics of technological diffusion. Blackwell, Oxford, UK.
[Ut94] Utterback, J. (1994) Mastering the Dynamics of Innovation. Harvard Business School Press, Cambridge MA, USA.
[ZS93] Zettelmeyer, F. and P. Stoneman (1993) Testing Alternative Models of new Product Diffusion. Economics of Innovation and New Technology, 2, 283-308.