2006

# Patents and Business Models for Software Firms

John R. Allison
John.Allison@mccombs.utexas.edu

Abe Dunn

Ronald J. Mann
*Columbia Law School*, rmann@law.columbia.edu

# THE UNIVERSITY OF TEXAS
# SCHOOL OF LAW

Law and Economics Research Paper No. 77
March 2006

*Patents and Business Models for Software Firms*

*Ronald J. Mann*
*Ben H. & Kitty King Powell Chair in Business and Commercial Law*
*Co-Director – Center for Law, Business & Economics*
*University of Texas at Austin, School of Law*

March 24, 2006 Draft

# PATENTS AND BUSINESS MODELS FOR SOFTWARE FIRMS

John R. Allison, Abe Dunn & Ronald J. Mann[*]

*We analyze the relation between patents and the different business models available to firms in the software industry. The paper builds on Cusumano's work defining the differences among firms that sell products, those that provide services, and the hybrid firms that fall between those polar categories. Combining data from five years of Software Magazine's Software 500 with data about the patenting practices of those software firms, we analyze the relation between the share of revenues derived from product sales and the firm's patenting practices. Accounting for size, R&D intensity, and sector-specific effects, the paper finds a robust positive correlation between product-based business models and patenting rates. We also present in this draft preliminary results suggesting that there is no significant relation between patenting practices and the extent to which the firm's revenues are derived from software products and services, as opposed to hardware or other lines of business.*

**1.** ***Introduction:*** If there is any single industry in which patents are controversial, it is the software industry. Patents have spread rapidly through the industry in recent years, shortly after doctrinal changes that have made it easier to obtain reliable patents on software-related inventions.[1] At the same time, there is a growing concern that the massive patent portfolios held by incumbent firms will chill innovation and entry in the industry.[2]

Still, empirical information about the role of software patents in the industry is scant. The most detailed published study of the industry is Graham and Mowery's 2003

---

[*] Allison is the Spence Centennial Professor in the Information, Risk, & Operations Management Department of the Red McCombs Business School, University of Texas. Dunn is a Ph.D. candidate in the Department of Economics, University of Texas. Mann is the Ben H. & Kitty King Powell Chair in Business and Commercial Law and Co-Director of the Center for Law, Business & Economics, at the University of Texas School of Law.

[1] See Ronald J. Mann, Do Patents Facilitate Financing in the Software Industry?, 83 Texas L. Rev. 961 (2005) [hereinafter Mann, Software Patents].

book chapter.[3] Their work studies patents in the listed UPC classes that are most commonly obtained by large software firms (primarily G06F). Comparing patent grants in those classes against R&D expenditures for large packaged-software firms, they conclude that the propensity to patent (measured in patents per $100M R&D dollars) rose by about 50% (from about 2.0 to 3.0) from 1988 to 1996. Regarding smaller firms, Mann's working paper with Sager discusses the role of patents in venture-backed software startups.[4] Generally, that paper shows that patents in the industry are used by only a minority of the 850 firms studied there (only 24% of the firms had patents at a date more than five years after their first financing). At the same time, patent acquisition is significantly correlated with any of several variables that are indicators of success of the firms. For example, 13% of the software firms with patents go public, while only 3% of those without patents go public. For present purposes, that work is important because it suggests that patents play a role of some importance in the development of firms seeking to enter the software industry, albeit one that depends substantially on the type of firm.

There has been a good deal of work, relying primarily on questionnaires, examining the value of patents in appropriating the profits of innovation in various industries. In general, that literature shows a broad spectrum of industries in which the effectiveness and importance of patents ranges from important and central to trivial and

---

[2] See Ronald J. Mann, Commercializing Open-Source Software: Do Property Rights Still Matter? (unpublished 2005 manuscript) [hereinafter Mann, Open Source].

[3] Stewart J. H. Graham and David C. Mowery, Intellectual Property Protection in the U.S. Software Industry, in Patents in the Knowledge-Based Economy 219 (Wesley M. Cohen & Stephen A. Merrill eds., 2003).

[4] Ronald J. Mann and Thomas W. Sager, Patents, Venture Capital, and Software Start-Ups (unpublished 2005 manuscript).

inconsequential.[5] There also has been a good deal of work, predominantly dealing with relatively large companies (for which data is more readily available) examining the relation between patent counts and R&D.[6]

There has been relatively little work, however, examining those problems on a firm-by-firm basis, within a particular industry, where the differences between particular types of firms can be captured on a micro-level. The only published study that examines the role of patents on that basis is the recent study by Hall and Ziedonis of the role of patents in the semiconductor industry.[7] Their paper studies a set of 100 large firms in the semiconductor industry, matching financial data with patent data to demonstrate that those firms have changed patenting behavior substantially between 1980 and 1995, with a substantial increase in patenting activity even after accounting for R&D expenditures and the size and types of firms.

---

[5] Richard C. Levin et al., Appropriating the Returns from Industrial Research and Development, 1987 Brookings Papers on Econ. Activity 783; Wesley M. Cohen et al., Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (Or Not) (NBER Working Paper Series, Working Paper 7552) (Feb. 2000), available at http://papers.nber.org/papers/w7552.

[6] E.g., Richard Blundell, Rachel Griffith & John van Reenen, Market Share, Market Value and Innovation in a Panel of British Manufacturing Firms, 66 Rev. Econ. Stud. 529 (1999); Michele Cincera, Patents, R&D, & Technological Spillovers at the Firm Level: Some Evidence from Econometric Count Models for Patent Data, J. Applied Econometrics 265 (1997); Bruno Crepon & Emmanuel Duguet, Research and Development, Competition, and Innovation Pseudo Maximum Likelihood and Simulated Maximum Likelihood Methods Applied to Count Data Models with Heterogeneity, 79 J. Econometrics 355 (1997); Bruno Crepon & Emmanuel Duguet, Estimating the Innovation Function from Patent Numbers: GMM on Count Panel Data, 12 J. Applied Econometrics 243 (1997); Jose G. Montalvo, GMM Estimation of Count Panel Data Models with Fixed Effects and Predetermined Instruments, 15 J Bus. & Econ. Stat. 82 (1997).

Although the literature provides no definitive theoretical framework for predicting when patents will be more and less useful, the most recent paper by Wesley Cohen and his co-authors takes steps toward a general explanation as part of a description of differences between the United States and Japan.[8] In their view, patents can play two distinct roles: as tools for exclusion (to be exploited through production within the patentholding firm), and as tools for licensing (to be exploited through licensing outside the boundary of the patentholding firm). They develop a distinction between "discrete" and "complex" products, finding evidence to support the idea that "complex" product industries in the United States rely more heavily on licensing to permit exploitation outside the boundaries of the firm.[9]

At first glance, it is difficult to apply that analysis to the software industry, because the software industry itself is strikingly heterogeneous. The Software 500 (a ranking of the top revenue grossing software firms), for example, includes more than 100 different sector designations within the industry. Moreover, what little we know about patenting in the software industry suggests every reason to believe that the role of patents differs substantially within the industry itself. For example, in Mann's recent working paper with Sager studying a dataset of venture-backed software firms, patenting rates

---

[7] Bronwyn H. Hall & Rosemarie Ham Ziedonis, The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979–1995, 32 Rand J. Econ. 101 (2001).

[8] Wesley M. Cohen, Akiro Goto, Akiya Nagata, Richard R. Nelson & John P. Walsh, R&D Spillovers, Patents and the Incentives to Innovate in Japan and the United States, 31 Research Pol'y 1349 (2002).

[9] Wesley M. Cohen, Akiro Goto, Akiya Nagata, Richard R. Nelson & John P. Walsh, *R&D Spillovers, Patents and the Incentives to Innovate in Japan and the United States*, Research Pol'y (forthcoming 2002).

differed substantially from sector to sector.[10] The dataset analyzed in that paper does not contain R&D information and thus is not well suited to analyzing the differences in propensity to patent on a firm-by-firm or sector-by-sector basis.

If we believe that the role of patents is important to the success of the firm as a profit-seeking enterprise, it would be natural for the utility of patents to relate to important distinctions in the business model that the firms pursues. Accordingly, in an effort to make some sense out of the bewildering array of software markets, we turned to the most prominent explication in the business-school literature of the distinctions in business models of software firms, Cusumano's continuum from products firms to services firms.[11] To simplify his complex analysis, products firms *generally* are characterized by higher operating margins, higher growth rates, and less stable market shares, whereas services firms *generally* have lower operating margins and lower growth rates, but can more readily establish stable market positions. From that perspective, the typical products firm (Microsoft, second in the current Software 500) is characterized by high-volume sales of non-customized products that customers can use "off the shelf" with little or no assistance. At the other end of the spectrum is the typical services firm (EDS, third in the current Software 500), which generates revenues by helping firms to install, design, and maintain software. In between is a large group of hybrid firms (like Oracle, eighth in the current Software 500). Those firms generally started by attempting to sell products, but later were forced by market conditions to provide ever-increasing levels of

---

[10] {See Table 3 in December 2005 Draft.}

[11] See Michael A. Cusumano, The Business of Software (2004).

customization, thus degrading their ability to sell high volumes of a pure high-margin product.

Although Cusumano does not emphasize the point, it is implicit in his analysis that the products model is *relatively* more effective for venture-backed startups than the services model. Because products firms can scale more easily than services firms – it is much easier to duplicate a product 10,000 times than the employees that provide services – successful products firms are more likely to produce the high returns venture capital investors seek.[12]

What is not clear from the existing literature is whether there is any relation between this distinction and the use of patents. The value of the products-services distinction in explaining other aspects of software business suggests that it might provide a useful lens for exploring the reasons for the apparent disparity of patenting practices in the industry. Moreover, a number of practical aspects of the industry suggested to us the likelihood that patents would be more useful for products firms than for services firms. For one thing, patents seem likely to be a relatively more effective tool for protecting innovation in products than in services. To the extent a firm can provide a unique level of skilled services, it may be feasible to maintain much of the differentiating knowledge in a tacit form, bound up with the skills of the individual employees.[13] Conversely, a products firm that sends its product out into the marketplace in many instances will be displaying its technology "near the surface" of the product, easily available for

[12] See Mann, Open Source, supra note 2.

appropriation by competitors.[14] If so, a patent that permits a firm to fence out competitors will have considerably more value to a products firm than to a services firm. This, in turn, suggests the hypothesis that products firms, because their technology is more difficult to protect than the technology of services firms, will produce more patents than services firms, all other things being equal.

**2.** ***Data:*** To investigate the role patents play in the software industry, we combined data from two separate sources. First, we collected data about the firms from *Software Magazine's* Software 500. Relying on questionnaires disseminated by the magazine, that list indicates the top 500 firms in the software industry each year by revenue. Anecdotally (based on interviews within the industry), we have the impression that the response rate is quite high. The list appears to be widely regarded as authoritative within the industry. Campbell-Kelly, for example, uses the list pervasively in his comprehensive history of the industry.[15] It is, for example, considerably more comprehensive than the Softletter 100 that Graham and Mowery use, which is limited to prepackaged software providers (and thus generally excludes services firms).

Because of considerable turnover in the industry, that list includes about 1000 firms for the five years. For each firm, the Software 500 list each year includes several

---

[13] See Ashish Arora, Andrea Fosfuri & Alfonso Gambardella, Markets for Technology: The Economics of Innovation and Corporate Strategy (2001) (discussing the importance of tacit knowledge).

[14] See Pamela Samuelson, Randall Davis, Mitchell D. Kapor & J.H. Reichman, A Manifesto Concerning the Legal Protection of Computer Programs, 94 Colum. L. Rev. 2308 (1994); Mann, Software Patents, supra note 1.

[15] Martin Campbell-Kelly, From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry (2003).

data points of interest, the total revenues, total revenues from software-related activities, % of revenues expended on research and development, number of employees, and % of revenues generated by the sale of services. Because the purpose of our study is to focus on firms that fairly can be characterized as software firms, we excluded the 18 firms that did not derive at least 20% of its total revenues from software in any of the five years for which we collected data.[16]

Of importance for our project, it extends from the largest firms in the industry (IBM and Microsoft were first and second throughout the five-year period) to quite small firms: the smallest firm in 2002, for example, was the firm of iCIMS, Inc., with annual revenues of only $400,000 (less than $35,000/month). This is important because previous similar empirical studies (such as the papers by Graham and Mowery about the software industry and the papers by Hall and Ziedonis about the semiconductor industry) rely on CompuStat data, which necessarily limits the analysis to relatively large firms. It also includes relatively recent information. Our study focuses entirely on activity beginning in the late 1990's, after the rise of the Internet and legal changes that arguably made it easier to obtain software patents. Thus, our paper is the first effort in the existing literature to provide any information at all about patenting practices in the modern software industry.

To quantify the patenting practices of the firms, we collected from Delphion a complete set of all of the 34,000 patents issued between January 1, 1998 and December

[16] The excluded firms are Cisco, Hitachi, Intel, NEC, Raytheon, Valassis, PreVision Marketing, VCON, Adaptec, Alstom ESCA, Amdahl, Brooktrout, Infolmage, International Network Services, Kasten Chase, MessageQuest, Template Software, and TYX.

31, 2002 to each of the firms listed in the Software 500. We then examined each of the 20,000 patents issued to firms other than IBM. {For the 14,000 IBM patents we read a random sample of about 300 patents and extrapolated from that sample.} The question was whether the patent, properly speaking, should be treated as a patent on a software invention.

Identifying a data set of software patents is a daunting task, to put it mildly. This is so for several reasons. First, there is no universally accepted definition of what a software patent is. Second, neither the U.S. Patent and Trademark Office (PTO) classification system nor the International Patent Classification (IPC) system was designed for such a purpose. Both systems focus on specific functions at a very low level of abstraction and are unsuitable for defining any technology area at a conceptual level. Third, even if these systems were suitable for identifying for defining a technology area, software is a critical element of inventions in so many disparate fields that it could not be adequately captured by a classification system.

To our knowledge, there have been only two significant efforts to identify a large data set of software patents. Graham and Mowery,[17] who did not attempt to define the term "software patent," used the IPC system in an effort to develop a data set of software patents owned by packaged software firms.[18] The IPC classes that they used do include large concentrations of patents on software inventions, but they also include substantial numbers of inventions that could not fall within anyone's definition of a software patent.

---

[17] Graham & Mowery, supra note 3.

Despite the inadequacy of the classification system for identifying software patents, the fact that they only included patents issued to packaged software firms means that their data set probably consisted almost exclusively of software patents. Thus, although their data set is probably not significantly overinclusive, it obviously is very underinclusive. In other words, if one wishes to have a data set that is representative of all software patents, their method would not work.

The other significant effort to identify a large set of software patents, by Bessen and Hunt,[19] offers a definition of the term "software patent" that includes, correctly in our view, patents on inventions in which the data processing algorithms are carried out by code either stored on a magnetic storage medium or embedded in chips ("firmware").[20]

---

[18] Graham & Mowery first identified packaged software firms and studied the IPCs of patents issued to those firms. They then selected those classifications to which large percentages of these firms' patents had been assigned. *Id*.

[19] JAMES BESSEN & ROBERT M. HUNT, AN EMPIRICAL LOOK AT SOFTWARE PATENTS 7-9. (Fed. Res. Bank of Philadelphia Working Paper No. 03-17, available at http://www.researchoninnovation.org/swpat.pdf.

[20] As Bessen and Hunt note, *id*. at 9, one of the current authors, John Allison, earlier employed a definition of software patent that excluded firmware, including only inventions in which the code implementing the data processing algorithms are stored on a magnetic storage medium. *See* John R. Allison & Mark A. Lemley, *Who's Patenting What? An Empirical Exploration of Patent Prosecution*, 53 VAND. L. REV. 2099, 2110-11 (2000); John R. Allison & Mark A. Lemley, *The Growing Complexity of the U.S. Patent System*, 82 B.U. L. REV. 77, 89 (2002); John R. Allison & Emerson H. Tiller, *The Business Method Patent Myth*, 18 BERKELEY TECH. L.J. 987, 1029 (2003). The reasons for using this definition were a combination of initial doubt and compromise with a coauthor, followed by a need for consistency. Each of those articles made use of the same data set of 1,000 randomly selected patents-in-general issued between mid-1996 and mid-1998. After a great deal more experience gained from closely reading thousands of computer-related patents, Allison became firmly convinced that the definition should include firmware. When he used the same set of 1,000 randomly selected patents in a subsequent article, he studied each patent again and reclassified them using a definition that included firmware. *See* John R. Allison, Mark A. Lemley, Kimberly A. Moore, & R. Derek Trunkey, Valuable Patents, 92 GEO. L.J. 435 (2004) (definition not explicitly

As we do, Bessen & Hunt reject the use of patent classifications for identifying a set of software patents.[21] Rather, Bessen and Hunt studied a random sample of patents, classified them according to their definition, and then developed a keyword search algorithm to identify a large data set of software patents.

We don't quarrel so much with the Bessen & Hunt definition as we do with the use of a keyword search. Allison's studies of thousands of computer-related patents convince him that the use of language in the titles, abstracts, written descriptions, and claims of patents, even in those dealing with the same area of technology, can be highly idiosyncratic among different patent owners. Moreover, software is a critical part of inventions in such far-flung fields that reliance on particular search terms will produce a data set that is substantially overinclusive and underinclusive at the same time.[22]

Turning to our methodology, we define a software patent as a patent in which at least one claim element consists of data processing, regardless of whether the code

---

provided in article). Allison used this more inclusive definition not only in this paper, but also in an ongoing study of university patents. Thus, when we say that identifying a large set of software patents is daunting, we speak from rich experience.

[21] Bessen & Hunt, *supra* note 19, at 10-11. The Bessen & Hunt definition of a software patent appears to include patents on inventions that "use" software as part of the invention, but excludes those that "use" off-the-shelf software:

> Our concept of software patent involves a logic algorithm for processing data that is implemented via stored instructions; that is, the logic is not "hard-wired." These instructions could reside on a disk or other storage medium or they could be stored in "firmware," that is, a read-only memory, as is typical of embedded software. But we want to exclude inventions that involve only off-the-shelf software—that is, the software must be at least novel in the sense of needing to be custom-coded, if not actually meeting the patent office standard for novelty.

*Id*. at 8.

[22] Bessen and Hunt identify substantial degrees of over- and underinclusiveness in the data set generated by their keyword search. *Id*. at 9.

carrying out that data processing is on a magnetic storage medium or embedded in a chip. After a great deal of experience, study, and thought, we have found that this is the only definition that is both appropriately inclusive and can be applied with principled consistency. Not only is it possible to apply the definition with consistency, but it also captures the realities of claim drafting. It is common for all or most of the elements in a patent claim to cover the prior art, with only one or perhaps two elements covering the purported novel and nonobvious advance. For example, computer hardware makers own large numbers of patents, the claims of which initially read as though they cover something like a generic router, printer, magnetic resonance imaging machine, or other hardware, when in fact the only purported novelty is in one element consisting of a function carried out by algorithms. Also, a claim covers the entire invention, and in a case like this the entire invention is not just the new algorithms in isolation but instead is a piece of hardware that allegedly does something different because of the new algorithms. Further, in the event of infringement in cases like this, one cannot calculate lost profits or reasonable royalties on the algorithms in isolation.

The most obvious problem with our methodology is that it requires reading every patent, an extraordinarily slow and laborious process. Although many patents are either obviously software patents or are obviously not software patents under this definition, there will always be a substantial percentage that must be studied with great care.[23] Claims are often quite obtuse, and in the computer field they are frequently rather broad,

---

[23] If one is studying a large population of patents from the computer-related industries, the percentage that must be carefully scrutinized is far higher than if one is studying a population of patents across a broad array of fields.

necessitating a close reading of not only independent but also dependent claims and a not uncommon resort to the written description to help interpret claim language. Another problem that simply cannot be avoided is that a degree of subjective judgment is occasionally required, which raises concerns about replicability. As a result, we do not claim that our data set of software patents is perfect, but we do contend that our error rate is extremely small, certainly far smaller than in any data set acquired by means of short-cuts such use of patent classifications or keyword searches.[24]

Using that methodology, about 68% (13,500) of the non-IBM patents qualified as software patents and about 55% of the IBM patents (extrapolating from the sample that we examined), for a blended total of about 62% (21,200) software patents. To provide additional data points for robustness checks (as described below), we subsequently collected a set of all of the patents issued to the firms from January 1, 2003 through June 30, 2005 (an additional 20,000 patents), but we did not analyze those patents to determine whether they were software patents or not.

### 3. *Analysis and Results*

o *Descriptive Statistics:* Because the existing empirical literature about patents in the software literature is so scant, even descriptive statistics about this dataset are interesting. We start with some basic descriptive statistics, set out in Table 1. {Because our statistical analysis focuses on the results of R&D activity in 1998 and 1999, the Table presents information about the firm and about the patents that arose from

---

[24] To the extent that there are any errors in our identification of a data set of software patents, those errors consist of not including a patent that should have been

applications made during those years.} As that Table shows, the patenting data are highly localized, with a mean of four patents and fourteen applications per firm, although more than three quarters of the firms show neither a patent nor an application.[25] At the same time, R&D per Employee displays a relatively normal distribution. This suggests on its face that there are important influences on patenting practices beyond the raw amount of investment in R&D.

**Table 1: Summary Statistics (1998-2000)**

| | |
|---|---|
| # of Observations | 1,114 |
| # of Firms | 646 |

| Variable Name | Mean | Standard Deviation | First Quartile | Median |
|---|---|---|---|---|
| Software Patents (For 1998-1999) | 4.0 | 32.3 | 0.0 | 0.0 |
| Patent Applications | 13.8 | 170.3 | 0.0 | 0.0 |
| R&D | 104.2 | 1721.1 | 3.1 | 8.4 |
| R&D per Employee | 0.033 | 0.027 | 0.020 | 0.029 |
| Employment (1000s) | 2154.7 | 14688.2 | 118.0 | 297.0 |
| Total Sales | 670.1 | 5109.5 | 14.6 | 40.8 |
| Fraction of Revenue from Services | 30.3 | 24.0 | 10.0 | 28.0 |
| Fraction of Revenues from Software | 88.5 | 20.1 | 84.4 | 100.0 |

To see this point more clearly, consider Table 2 below, which displays detailed information about the distribution of firms that received patents based on applications filed during the period of data collection.

included. Again, however, we believe that any errors of underinclusiveness in our data set are extremely small.

[25] Our patent and application data is necessarily truncated. As discussed below, our analysis associates each patent with the year in which the application was filed. Even now, there remains the possibility that patents will issue for which applications were filed in 1998 and 1999. For reasons explained below, we do not believe that possibility undermines the robustness of our analysis.

**Table 2: Distribution of Patent Portfolios**

| | Software Patents 1998 and 1999 | All patents for the years 1998-2000 |
|---|---|---|
| # of Patents | # of Obs. | # of Obs. |
| 0 | 596 | 899 |
| 1 | 49 | 90 |
| 2 | 14 | 30 |
| 3 | 5 | 17 |
| 4 | 6 | 16 |
| 5 | 0 | 3 |
| 6 | 1 | 2 |
| 7 | 4 | 2 |
| 8 | 0 | 3 |
| 9 | 1 | 2 |
| 10 or more | 27 | 50 |
| Total # Obs. | 703 | 1114 |

Our analysis also incorporates information on firm sectors. There are widely heterogeneous sectors in the software industry that may also explain some of the differences in patenting practices between firms. Because the Software 500 uses more than 100 different sectoral designations, many of the sectors include very few firms. Accordingly, we constructed a modified set of sectoral designations, which consolidates the Software 500's designations into "only" 36 sectors. The table below shows the different sectors within the software industry and some basic descriptive statistics for firms in each sector.

Table 3: **Software Sector Descriptive Statistics (1998 and 1999)**

| | R&D | Empl. (1000s) | R&D per Emp | Services | SW Patents | SW Patents | SW Patents | SW Patents | Count of Obs. | Count of Firms |
|---|---|---|---|---|---|---|---|---|---|---|
| | Median | Median | Median | Median | Mean | Median | 75% | 90% | | |
| Application Dev'l'pm't | 6.95 | 220 | 0.032 | 22 | 0.23 | 0 | 0 | 1 | 61 | 42 |
| Application Serv. Prov. | 1.39 | 69.5 | 0.017 | 26 | 1.75 | 0 | 3.5 | 7 | 4 | 4 |
| Asset/Technol. Mgmt. | 7.38 | 421.5 | 0.027 | 38.5 | 0.00 | 0 | 0 | 0 | 12 | 7 |
| Business Intelligence | 12.83 | 446 | 0.029 | 28 | 0.37 | 0 | 1 | 1 | 27 | 20 |
| Bus. Process Mgmt. | 21.51 | 525 | 0.035 | 54 | 0.33 | 0 | 1 | 1 | 3 | 3 |
| Computer Ass'd Draft'g | 5.72 | 283 | 0.020 | 32 | 1.57 | 0 | 0 | 11 | 7 | 5 |
| Content/Doc. Mgmt. | 8.71 | 188.5 | 0.031 | 18 | 0.18 | 0 | 0 | 0 | 28 | 22 |
| Collab./Proj. Mgmt. | 10.40 | 376 | 0.029 | 38 | 0.00 | 0 | 0 | 0 | 4 | 2 |
| Cust. Rel'nship Mgmt. | 5.84 | 206 | 0.031 | 35 | 0.32 | 0 | 0 | 0 | 41 | 32 |
| Database | 18.84 | 505.5 | 0.032 | 25 | 9.69 | 0 | 2 | 62 | 16 | 12 |
| Disaster Recovery | 0.84 | 38 | 0.022 | 27 | 0.00 | 0 | 0 | 0 | 1 | 1 |
| Data Warehouse | 298.02 | 5679 | 0.044 | 12 | 15.29 | 0 | 49 | 57 | 7 | 5 |
| Enterprise Appl. Integr'n | 6.48 | 261 | 0.030 | 30.5 | 0.06 | 0 | 0 | 0 | 36 | 26 |
| E-Business Applications | 7.27 | 217.5 | 0.027 | 31 | 0.25 | 0 | 0 | 1 | 44 | 31 |
| E-Commerce | 13.57 | 195 | 0.032 | 13 | 0.00 | 0 | 0 | 0 | 5 | 5 |
| E-Learning | 34.18 | 1500 | 0.023 | 26 | 0.00 | 0 | 0 | 0 | 3 | 3 |
| Enterprise Res. Planninc | 11.25 | 522.5 | 0.023 | 40.5 | 0.10 | 0 | 0 | 0 | 62 | 44 |
| Financial Applications | 8.29 | 270 | 0.027 | 32 | 0.02 | 0 | 0 | 0 | 42 | 29 |
| Geogr. Info. Systems | 18.05 | 420 | 0.029 | 9 | 6.00 | 0 | 9 | 32 | 7 | 5 |
| Healthcare | 3.33 | 106 | 0.020 | 22 | 0.00 | 0 | 0 | 0 | 9 | 6 |
| Human Resources | 3.05 | 31 | 0.022 | 26.5 | 0.00 | 0 | 0 | 0 | 16 | 10 |
| Infrastructure | 7.85 | 250 | 0.037 | 23 | 1.33 | 0 | 0 | 2 | 55 | 39 |
| IT Sourcing | 5.73 | 300 | 0.021 | 53 | 40.15 | 0 | 0 | 138 | 13 | 9 |
| Marketing Automation | 2.23 | 78 | 0.024 | 23 | 0.00 | 0 | 0 | 0 | 5 | 3 |
| Middleware | 5.67 | 200 | 0.029 | 17 | 0.42 | 0 | 1 | 2 | 19 | 16 |
| Operating Systems | 25.71 | 641 | 0.039 | 14.5 | 43.14 | 1 | 2 | 271 | 14 | 10 |
| Portal Tools | 3.32 | 151.5 | 0.026 | 34.5 | 0.00 | 0 | 0 | 0 | 8 | 7 |
| Publishing/Graphics | 9.12 | 300 | 0.034 | 0 | 3.20 | 1 | 4 | 11 | 5 | 5 |
| Retail Applications | 27.74 | 1068 | 0.026 | 74 | 0.00 | 0 | 0 | 0 | 1 | 1 |
| Supply Chain | 8.25 | 253 | 0.030 | 42.5 | 0.63 | 0 | 0 | 1 | 54 | 38 |
| Security | 12.11 | 450 | 0.032 | 13.5 | 1.50 | 0 | 1.5 | 6.5 | 20 | 15 |
| Sales Force Automation | 7.21 | 381 | 0.039 | 39.5 | 0.00 | 0 | 0 | 0 | 4 | 4 |
| System Integration Serv: | 9.53 | 2240 | 0.011 | 49 | 65.00 | 0 | 35.5 | 361 | 16 | 10 |
| Storage Management | 3.79 | 135 | 0.032 | 11 | 7.14 | 0 | 4 | 43 | 7 | 5 |
| Vertical Indus. Appl. | 3.57 | 273.5 | 0.015 | 39.5 | 1.60 | 0 | 0 | 0 | 20 | 15 |
| Wireless/Mobile | 3.33 | 119.5 | 0.033 | 37 | 0.25 | 0 | 0.5 | 1 | 4 | 3 |
| Other | 6.04 | 164 | 0.031 | 20 | 1.65 | 0 | 0 | 2 | 23 | 17 |
| Total # of Firms: | | | | | | | | | | 511 |

The table shows the median of size and service revenue to show the heterogeneity of the typical market. For example, the median data warehousing firm has about 5,000 employees, while the median disaster recovery firm has only 38. Similarly, the typical data warehousing firm derives only 12% of its revenues from services, while the median retail applications firm derives 74% of its revenues from services. Of particular purport for our work is the great variation in patenting practices, with quite a number of reasonably well-populated sectors entirely devoid of patents (human resources software, for example), and others in which substantial portfolios exist (operating systems and systems integration services, for example, with an average of more than 40 patents per firm).

o *The Model:* Because the basic purpose of our inquiry is to understand the relation between the business model of a particular firm and the patenting practices displayed in Table 2, we proceeded to construct a patent production function, generally following the methodology of Hall and Ziedonis. The output of this production function is the number of patents applied for and successfully obtained by a firm. The dependent variable naturally takes on the value of zero and positive integer values (i.e. 0,1,2,3…). Typical models used to examine this type of data are count models that include the commonly used Poisson and Negative Binomial models. These models have been applied in several papers to examine patent production. Papers using this approach include Hausman, Hall & Griliches (1984)[26] and Hall and Ziedonis (2001)[27] that examine

---

[26] Jerry Hausman, Bronwyn H. Hall & Zvi Griliches, Econometric Models for Count Data with an Application to the Patents R&D Relationship, 52 Econometrica 909 (1984).

patent production in the semiconductor industry, and Bessen and Hunt (2001) examining software patents. We discuss in detail below how we have addressed the problems in matching the distributional assumptions of those models to the characteristics of this dataset.

Similar to previous work we assume that the number of patents applied for in a year is a function of a firm's R&D spending and other characteristics of the firm. The subscript $i$ denotes the firm, and the subscript $t$ denotes the year. The number of patents produced by firm $i$ at time $t$ is denoted by the variable $y_{it}$. We assume that the number of patents is a function of observable and unobservable factors. The primary estimates in this paper assume that the unobserved component has a Poisson distribution. Under the Poisson distribution assumption the expectation of $y_{it}$ takes the form:

(1) $E(y_{it}) = \exp(x_{it}\beta)$

The expectation of the model is a function of observed exogenous variables $x_{it}$ and a vector of parameters $\beta$. The parameters of the model are estimated using maximum likelihood. We note here an important feature of our analysis. In general, a maximum likelihood model will not be consistent unless the distributional assumption of the model is correct. However, Gourierox, Monfront & Trognon (1984)[28] show that if the mean of the above equation is correctly specified then the estimate of $\beta$ will be consistent even if the data rejects the Poisson distributional assumption. The standard

[27] Hall & Ziedonis, supra note 7.

[28] Christian S. Gourieroux, Alain Montfort & Alain Trognon, Pseudo Maximum Likelihood Methods: Application to Poisson Models, 52 Econometrica 701 (1984).

errors must be corrected to be robust to alternative distributions. This is important because the assumption that the variance of the Poisson model is equal to the mean is restrictive and often (as with the data here) incorrect in practice, typically when the excess of the variance over the mean reflects "over-dispersion."

- *Variables:* The variables analyzed here include patent counts for firms and exogenous variables explaining those patents. All the variables examined in this paper are specific to a particular firm in a year. This paper explores two measures of patent output.

One measure includes a count of the number of software patents applied for and received by the firm in the year. Each of the 54,000 patents was allocated to the year in which the patent application was filed. Because there is a lag in the time between a patent is applied for and received, we can be sure that the July 2005 searches did not locate all of the patents attributable to applications for any year in this dataset (the earliest year being 1998). Nevertheless, the information the dataset reveals about the time of patent examination allows us to assess the extent of the truncation problem. Thus, it appears from the data that the median number of years it takes to have a software application approved is 2.47 years and the 75$^{th}$ percentile is 3.08 years. This suggested that the truncation issues for 2000-2002 would be quite serious, but that those issues would not be serious for 1998 and 1999 – the fastest granted 1999 application we could have missed would have undergone more than five and one-half years of examination and the fastest 1998 application truncated by our search would have languished in the PTO for six and one-half years. Accordingly, the analysis presented below relies only on the 2833 software patents issued with respect to 1998 and 1999 applications.

A second measure includes a count of the total number of patents including the number of software and other patents received with respect to applications filed in a particular year. As discussed above, this rests on searches to locate patents issued from January 1, 1998 through June 30, 2005. Because we have collected an additional year of total patent information, we analyze three years of total patent data (1998, 1999, and 2000), which involves a total of 15,420 patents.

The primary explanatory variables relate to the lines of business of the firm. We have two such variables: *Services* and *Fraction of Software Sales*. The *Services* variable is the fraction of software revenues that are from services multiplied by 100. This variable enters the model linearly. The second business line variable, *Fraction of Software Sales*, is the software sales for the current year, divided by total revenues, multiplied by 100. This variable is a proxy for how much of the firm is concentrated on software. Because many of the largest software firms are predominantly engaged in the sale of hardware (firms like Hewlett-Packard and Sun), this variable might capture differences in patenting practices between hardware firms and software firms. Both the *Fraction of Software Sales* variable and the *Services* variables are expressed as percentages falling in the range between 0 and 100.

We also use a number of other control variables. The most important variable explaining the number of patents produced is *R&D Expenditure* in 2002 dollars. This is a key explanatory variable in the estimations in our paper, as it has been in previous papers such as Hall & Ziedonis (2001). A second major control variable is the size of the firm measured by the number of employees in thousands, *Employee*. In addition, the intensity of R&D investment for the size of the firm is also used (*R&D/Employee*). These variables enter the production function in the log form including, *log(Employee), log(R&D)*, and *log(R&D/Employee)*. We use the log form because of the large differences in the size of firms observed in the data. The log transformation captures the skewed nature of the variables in the sample, and allows for interpretation of coefficients as constant elasticities. The *R&D* variable takes on a zero value for some observations, but the log of zero is undefined. To account for this problem, we include a variable *Zero R&D* that is 1 when *R&D*=0 and 0 otherwise. The variables *log(R&D)* and *log(R&D/Employee)* are set equal to zero in the cases where *R&D*=0. Firms observed as having 0 *R&D* represent less than 1% of the data.

Another set of control variables are year dummies, which account for shifts in patenting rates over the years.

- *Results Using Software Patents:*

Table 4 below summarizes the results of analysis using software patents as the dependent variable. Because the *Services* variable is a fraction of software revenues, and because our hypothesis is that the devotion of the firm to a products model should relate positively to the firm's propensity to patent innovations related to software, this model should provide the clearest test of our primary hypothesis. We report t-statistics in parentheses after the coefficient.

Table 4: **Propensity to Produce Software Patents**

| | (1) Poisson | | (2) Negative Binomial | | (3) Poisson | | (4) Negative Binomial | |
|---|---|---|---|---|---|---|---|---|
| Log(R&D/Employee) | 0.889 | (3.83) | 0.828 | (3.28) | 0.599 | (2.56) | 0.934 | (3.44) |
| Log(Employee) | 1.171 | (14.86) | 1.094 | (13.84) | 1.138 | (12.07) | 1.061 | (12.09) |
| Services | -0.023 | -(2.87) | -0.040 | -(5.58) | -0.015 | -(2.11) | -0.020 | -(2.43) |
| Fraction Software Sales | -0.009 | -(1.78) | -0.002 | -(0.25) | -0.008 | -(2.43) | -0.005 | -(0.65) |
| Zero R&D | -4.770 | -(5.11) | -4.259 | -(2.33) | -4.020 | -(5.07) | -4.907 | -(2.38) |
| Year 1999 | 0.009 | (0.04) | -0.096 | -(0.31) | -0.136 | -(0.54) | -0.225 | -(0.70) |
| Constant | -3.788 | -(2.42) | -3.555 | -(2.84) | 1.349 | -(0.23) | -3.797 | -(2.74) |
| Alpha | | | 4.458 | (6.25) | | | 2.921 | (5.67) |
| Sector Fixed Effects | No | | No | | Yes | | Yes | |
| # Observations | 703 | | 703 | | 612 | | 612 | |
| # Firms | 511 | | 511 | | 445 | | 445 | |
| log-likelihood | -1049 | | -500.2 | | -749.8 | | -448.7 | |

*Poisson:* The first column is the baseline Poisson specification. The first test of the main hypotheses is based on the coefficient on the *Services* variable in the first column of the table. The estimates show the coefficient on the service variable to be both negative and statistically significant, providing supporting evidence for the main hypothesis. Therefore, holding all other factors fixed, an increase in the fraction of revenues from software services implies fewer patents are produced. The impact of the *Services* variable on the number of patents shows that a 1% increase in the percent of software sales coming from services (e.g. percent of sales increasing from 50% to 51%), implies a 2.3% decrease in the number of patents produced. A more extreme result suggests that the magnitude of the *Services* variable is also economically significant. A firm that derives all its revenues from products (e.g. *Service* =0%) is expected to produce 230% more patents than a firm entirely devoted to providing services (e.g. *Service*=100%).

The variables capturing R&D intensity, *log(R&D/Employee)* and those capturing firm size, *log(Employee)*, are of particular interest to examine and compare to other results in the literature. The coefficients on these variables may be interpreted as constant elasticities. For instance, all other things held fixed, the coefficient of 1.17 on *log(Employee)* implies that a 10% increase in the number of employees causes an 11.7% increase in the number of software patents. This result is slightly higher, but comparable to other results found in the literature, including Hall and Ziedonis (2001) that find a coefficient of .989 in the semi-conductor industry and Bessen and Hunt (2004) who find a coefficient of .88 in the production of software patents by firms that are not necessarily in the software industry. Together with the other results, this suggests that returns to scale in number of employees is approximately constant in the software industry. In other words, the firms patent in proportion to their size, so that a doubling in the size of a firm is predicted to cause a doubling in the number of patents produced. The elasticity of R&D intensity on patenting is .89. Again, this is similar to the results found by Bessen and Hunt (2004) of 1.01.[29] However, these estimates are much larger than results in the semiconductor industry of .18 found by Hall & Ziedonis (2001). Generally, this suggests that the effects of size in the software industry are about the same as those in the semi-conductor industry, but that the effect of R&D intensity on software patenting is quite a bit greater than its effect on semi-conductor patenting. This may reflect the importance

[29] Bessen and Hunt (2001) find different results when accounting for firm level heterogeneity using fixed effects. However, their fixed effect estimation excludes firms with zero patents. Such firms include a majority of firms in this paper and in the Bessen and Hunt paper. Therefore, we compare the basic Poisson regressions as these include all firms and are less prone to sample selection bias.

of people as an input into innovation in the software industry relative to the semiconductor industry.

The other variables used in the estimation help control for various other factors that affect patent production. For example *Fraction of Software Sales* accounts for the possibility (suggested in interviews reported in Mann (2005)) that patenting practices will vary depending on how much the firm is devoted to selling software products – because hardware firms will patent more than software firms. Although this variable is sometimes statistically insignificant, and the coefficients are quite small compared to the coefficients on the variables discussed above, these estimates do tend to suggest what was suggested in those interviews: the estimates suggest the possibility that firms that have a larger fraction of revenue coming from software and other products produce less patents.

Another important control variable is a year dummy variable. The dummy variable accounts for shifts in patenting practices over the years, and also helps account for any remaining truncation that may limit the number of patents observed in 1999 relative to 1998. The low coefficient and t-statistic for that variable support our view that truncation problems are relatively minor: if truncation were affecting our results, we would expect 1998 and 1999 to differ in some noticeable way, because truncation for 1999 is likely to be greater than for 1998.

Finally, as one might expect, the control variable *Zero R&D* is found to be negative and significant.

*Negative Binomial:* Although the goodness of fit test rejects the Poisson distributional assumption, we nevertheless report the results of this analysis, following Gourierox, Monfront & Trognon (1984), discussed above. As recommended there, we use heteroscedastic-consistent standard errors to calculate t-statistics.[30] Still, even though the Poisson estimates are consistent, they are less efficient than a maximum likelihood model with a correctly specified distribution. One model typically used when overdispersion is present is the Negative Binomial model. The Negative Binomial model is consistent only if the true distribution is Negative Binomial; however, if this is the true specification then the estimate is more efficient than the Poisson model. The second model in Table 4 shows estimates from the Negative Binomial II model. The parameter alpha is the overdispersion parameter. Alpha is significantly different from zero suggesting overdispersion is present in the Negative Binomial model. The results found using the Negative Binomial model do not change the key results. The coefficients and t-statistics on firm size and R&D intensity are close to those found using the Poisson model, and the magnitude of the service coefficient remains statistically significant. In fact, the coefficient on the service variable increases to .399, which is nearly double the effect found using the Poisson model.

*Fixed Effects Models:* Columns (3) and (4) include sector-specific fixed effects that control for differences in the propensity to patent across different sectors in the software industry. Inclusion of the sector fixed effects is important for two reasons. First, from the previous estimates it is unclear whether the effect the service variable is capturing different propensities to patent because of differences in the services across sector, or whether the product/services distinction is also important within sectors.

---

[30] The goodness of fit test is based on the deviance statistic. The standard error estimates used to compute the t-statistics are robust to heteroskedasticity and misspecification of the distribution. To account for the multiple observations of some firms and the consequent possibility of autocorrelation, the standard errors are clustered.

Inclusion of the sector fixed effects along with the service variable focuses the test of the service variable. Specifically, this model tests the hypothesis of whether the product/services distinction is important within sectors. The second reason to include sector fixed effects is that a key assumption for estimates to be consistent is that the error term $\varepsilon_{ijt}$ be uncorrelated with the exogenous variables $x_{ijt}$. If there are unobserved characteristics in the sector that are correlated with $x_{ijt}$ then both the Poisson and Negative Binomial models from column (1) and (2) are inconsistent. Including sector specific fixed effects corrects for this type of inconsistency by allowing for correlation between sector specific unobservables and $x_{ijt}$.

Inclusion of sector specific fixed effect necessitates dropping several observations from the analysis. Sectors that have no patents are excluded from the analysis because the sector specific fixed effects entirely explain the number of patents in those sectors. In addition, the sector category marked "Other" is also excluded because it does not represent any particular sector.[31] The sector fixed effect estimates are based on the remaining 612 observations from the 445 remaining firms. We test the joint statistical significance of the sector fixed effects by using a likelihood ratio test based on the selected sample with 612 observations. For both models, we reject the null hypothesis that the sector specific fixed effects have no explanatory power at the 95% confidence level.

The results of the Poisson model and the Negative Binomial show that the *Services* variable continues to be negative and statistically significant. What this suggests

[31] The results do not change significantly if the "Other" category is included in the estimates.

is that the devotion of a firm to a products or services model is important, even within a particular sector. Thus, the data do not suggest simply that some sectors of the industry rely more on products and some more on services, or that levels of patenting activity reflect those differences. The data suggest that there are important differences along the products/services continuum, even *within* particular sectors, and that differences even at that specific level relate to differences in patenting activity. These results suggest that the product/services distinction is important within software sectors. To be sure, the magnitude of the coefficients on *Services* does drop considerably (from .023 and .040 to .015 and .020, respectively), but this apparently suggests that the sectoral differences capture a portion of the difference in patenting activity.[32]

*Lagged Data:* The above estimates use the explanatory variables from the same year as the dependent variable. However, one might be concerned that there is actually a lagged effect of services or other explanatory variables on patenting. Table 5 below provides both the Poisson and Negative Binomial estimates of the number of patents from 1999 based on lagged explanatory variables observed in 1998. Due to limitations in the data, there were only 192 observations to perform this robustness check. The results show that the *Services* variable continues to have a negative coefficient in both estimates, though it loses its significance in the Poisson model. {Table 8 in the next section analyzes lagged data involving total patents, for which we have more data points.}

[32] The drop is not caused by a change in the sample from specifications (1) and (2) to specifications (3) and (4). Runs that we do not report show approximately the same magnitude in the *Services* variable if we use specifications (1) and (2) for the 612 observations available for specifications (3) and (4).

Table 5: **Propensity to Produce Software Patents**

| Lagged Variables | (1) Poisson | | (2) Negative Binomial | |
|---|---|---|---|---|
| Log(R&D/Employee) | 0.196 | (0.58) | -0.048 | -(0.08) |
| Log(Employee) | 1.254 | (8.24) | 1.290 | (6.22) |
| Services | -0.014 | -(0.89) | -0.087 | -(2.98) |
| Fraction Software Sales | -0.006 | -(0.83) | 0.008 | (0.47) |
| Zero R&D | 0.256 | (0.18) | 1.010 | (0.34) |
| Year 1999 | - | - | - | - |
| Constant | -7.982 | -(3.04) | -7.707 | -(2.41) |
| Alpha | | | 5.934 | (3.28) |
| Sector Fixed Effects | No | | No | |
| # Observations | 192 | | 192 | |
| # Firms | 192 | | 192 | |
| log-likelihood | -251.6 | | -126.4 | |

- *Results Using All Patents:* We also analyzed a model using total patents as the dependent variable. This model has the advantage that we have more data points (because we have collected an additional year of patent data that have not been broken down into software and non-software patents). Thus, we analyze three years of data (1998, 1999, and 2000) rather than the two years analyzed in the previous section. It also has the advantage that it provides a more accurate test of our second hypothesis (related to *Fraction of Software Sales)*. On the other hand, it has the disadvantage that it is tied less precisely to our primary hypothesis related to *Services*, so with respect to that variable it serves primarily as a robustness check.

One problem that arises in using the total number of patents is that there are more outlier observations for the dependent variable on total number of patents produced relative to the number of software patents. Specifically, there are 9 observations (for five firms) where a firm has over 500 patents,[33] while the remaining observations have 200 or fewer patents. The approach taken to deal with these outliers is to analyze the data both with and without the outliers. Tables 6 and 7 shown below include the same analysis as table 4, but with the dependent variable of total patents by the firm with respect to applications from a particular year. Table 6 includes the outliers, while Table 7 excludes them. The most obvious effect of including the outliers is to decrease the coefficient on R&D intensity, *Log(R&D/Employee).* Also, the outliers seem to affect coefficient estimates in the Poisson model, but have little effect on the Negative Binomial model. A likely reason for the outliers affecting the Poisson model is that there is no error term to account for the additional dispersion in the error term caused by including the outliers. The remainder of this section focuses on estimates that exclude the 9 outliers. Excluding the 9 outliers, the impact of both patenting intensity and firm size are similar to the estimates using the number of software patents as the dependent variable. Note that the key hypothesis predicting a negative coefficient on the variable *Services* holds whether or not the outliers are excluded.

[33] The outliers are IBM, Hewlett-Packard, Sun, Microsoft, and Compaq.

Table 6: **Propensity to Produce Patents exlcluding Outliers**

| | (1) Poisson | | (2) Negative Binomial | | (3) Poisson | | (4) Negative Binomial | |
|---|---|---|---|---|---|---|---|---|
| Log(R&D/Employee) | 0.614 | (2.31) | 0.915 | (4.81) | 0.681 | (2.70) | 0.800 | (4.53) |
| Log(Employee) | 1.071 | (13.88) | 1.046 | (13.79) | 1.143 | (14.43) | 1.055 | (14.41) |
| Services | -0.026 | -(3.80) | -0.033 | -(5.27) | -0.015 | -(2.28) | -0.025 | -(4.63) |
| Fraction Software Sales | -0.008 | -(2.13) | -0.004 | -(0.78) | -0.008 | -(1.60) | -0.004 | -(0.94) |
| Zero R&D | -4.219 | -(4.08) | -5.157 | -(6.05) | -4.001 | -(3.89) | -4.148 | -(4.85) |
| Year 1999 | 0.442 | (1.49) | 0.479 | (1.53) | 0.527 | (2.26) | 0.443 | (1.87) |
| Year 2000 | -0.121 | -(0.38) | -0.179 | -(0.53) | 0.116 | (0.52) | -0.221 | -(0.91) |
| Constant | -3.770 | -(3.69) | -2.869 | -(3.67) | -5.376 | -(4.33) | -4.333 | -(4.09) |
| Alpha | | | 4.187 | (5.89) | | | 3.135 | (5.89) |
| Sector Fixed Effects | No | | No | | Yes | | Yes | |
| # Observations | 1105 | | 1105 | | 1029 | | 1029 | |
| # Firms | 642 | | 642 | | 589 | | 589 | |
| log-likelihood | -1975 | | -905.2 | | -1427 | | -830 | |

Table 7: **Propensity to Produce Patents including Outliers**

| | (1) Poisson | | (2) Negative Binomial | | (3) Poisson | | (4) Negative Binomial | |
|---|---|---|---|---|---|---|---|---|
| Log(R&D/Employee) | 0.275 | (2.16) | 0.954 | (5.19) | 0.120 | (1.24) | 0.868 | (5.21) |
| Log(Employee) | 1.472 | (16.55) | 1.115 | (16.07) | 1.513 | (11.95) | 1.123 | (17.21) |
| Services | -0.023 | -(3.67) | -0.033 | -(5.19) | -0.011 | -(1.47) | -0.024 | -(4.53) |
| Fraction Software Sales | 0.002 | (0.45) | -0.007 | -(1.42) | 0.000 | (0.17) | -0.007 | -(1.66) |
| Zero R&D | -2.870 | -(5.00) | -5.472 | -(6.85) | -1.915 | -(3.00) | -4.807 | -(5.97) |
| Year 1999 | 0.105 | (0.32) | 0.492 | (1.50) | -0.350 | -(0.88) | 0.441 | (1.82) |
| Year 2000 | 0.043 | (0.14) | -0.129 | -(0.37) | -0.462 | -(1.12) | -0.197 | -(0.81) |
| Constant | -8.942 | -(7.30) | -2.925 | -(3.64) | -10.749 | -(9.25) | -4.276 | -(4.06) |
| Alpha | | | 3.999 | (5.69) | | | 2.996 | (5.81) |
| Sector Fixed Effects | No | | No | | Yes | | Yes | |
| # Observations | 1114 | | 1114 | | 1038 | | 1038 | |
| # Firms | 646 | | 646 | | 593 | | 593 | |
| log-likelihood | -3570 | | -988.7 | | -2098.4 | | -911.8 | |

The second hypothesis examined in this paper is that the fraction of sales from software products or services will have some effect on the propensity to patent. To test this hypothesis we look at the coefficient on the variable *Fraction of Software Sales*. Because the hypothesis here relates to the overall patenting philosophy of the firm, the hypothesis is tested more directly with the data in this section of the paper as the dependent variable (total patents) rather than the more limited data on software patents analyzed above. The results, however, are similar to those for the software patents data. Thus, the coefficient on *Fraction of Software Sales* is quite small and almost always negative, but now the coefficient attains occasional significance. Therefore, there is some evidence that more software-oriented firms patent less. However, further evidence is necessary to determine whether this result holds. A key limitation in testing this hypothesis is that the *Fraction of Software Sales* has a limited range between 80% and 100% because all firms that have a *Fraction of Software Sales* variable less than 80% are dropped. Our skepticism in the robustness of the results related to this variable is bolstered by the occasional positive coefficients (particularly in Table 8 discussed below).

Finally, Tables 8A and 8B shown below include a number of additional robustness checks. The first columns reflect a Random Effect Poisson estimate that includes sector-specific fixed effects. This model follows the work by Hausman, Hall and Griliches (1984). The Random Effect Poisson model assume that there is a firm specific error term that is independent of the $x_{ijt}$ and has a Gamma distribution. The coefficient on the *Services* variable is both negative and significant, supporting the hypothesis distinguishing product and service firms. The estimates also show that the coefficient on *Fraction of Software Sales* is both positive and insignificant.

The second specification includes firm-level fixed effect estimates. This model is desirable because it accounts for potentially important firm-specific heterogeneity. However, about three-quarters of the sample observations must be dropped to estimate this model, because it must exclude firms that do not patent. Therefore, the fixed effect estimates eliminate the potential bias caused by firm-specific error correlated with exogenous variables, $x_{ijt}$, but introduces sample selection bias caused by dropping approximately 3/4 of the relevant sample. Again, as in all of our specifications, the estimates show a negative coefficient on the *Services* variable, but it has lost its significance with the truncation of the sample. Supporting our skepticism related to the *Fraction of Software Services* variable, the estimates show a positive (and insignificant) coefficient on that variable.

The remaining specifications in table 8A, (3)-(6), follow the estimates previously examined, but following the specifications in Table 5 these estimates only include lagged exogenous variables. The estimates support much of the previous findings. The coefficient on the *Services* variable is negative and significant for all estimates, and the sign and significance of the coefficient on *Fraction of Software Sales* varies depending on the model. The estimates also show that R&D intensity and firm size are comparable to magnitudes found in previous estimates, with the exception that the impact of R&D intensity on the propensity to patent in the Negative Binomial models is noticeably higher in these lagged specifications.

Table 8: **Propensity to Produce Patents without Outliers**

| | (1) Poisson | | (2) Poisson | | (3) Poisson | | (4) Neg. Binomial | | (5) Poisson | | (6) Neg. Binomial | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Log(R&D/Employee) | 0.392 | (3.24) | 0.131 | (0.46) | 0.656 | (2.02) | 1.167 | (3.62) | 0.613 | (1.72) | 1.230 | (4.45) |
| Log(Employee) | 1.008 | (13.59) | 0.581 | (1.84) | 1.025 | (11.28) | 1.165 | (11.04) | 1.250 | (9.78) | 1.234 | (13.74) |
| Services | -0.010 | -(4.21) | -0.007 | -(1.12) | -0.019 | -(2.30) | -0.049 | -(5.69) | -0.028 | -(2.39) | -0.054 | -(5.61) |
| Fraction Software Sales | 0.002 | (0.88) | 0.001 | (0.12) | -0.008 | -(1.51) | 0.000 | (0.02) | -0.013 | -(2.09) | 0.002 | (0.25) |
| Zero R&D | -2.494 | -(1.80) | -1.046 | -(0.09) | -4.515 | -(2.53) | -5.012 | -(3.44) | -2.909 | -(1.73) | -5.068 | -(4.13) |
| Year 1999 | 0.088 | (1.00) | -0.007 | -(0.02) | - | - | - | - | - | - | - | - |
| Year 2000 | -0.358 | -(3.34) | -0.373 | -(1.09) | 0.323 | (1.40) | 0.638 | (1.75) | 0.078 | (0.35) | 0.457 | (1.53) |
| Constant | -6.250 | -(7.83) | | | -3.287 | -(2.52) | -2.754 | -(1.85) | -6.014 | -(3.20) | -4.070 | -(2.75) |
| | | | | | | | | | | | | |
| Alpha | 3.282 | (7.26) | | | | | 3.717 | (5.16) | | | 2.718 | (5.15) |
| | | | | | | | | | | | | |
| Sector Fixed Effects | Yes | | No | | No | | No | | Yes | | Yes | |
| | | | | | | | | | | | | |
| Firm Fixed Effects | No | | Yes | | No | | No | | No | | No | |
| | | | | | | | | | | | | |
| Random Effects | Yes | | No | | No | | No | | No | | No | |
| | | | | | | | | | | | | |
| Lag Variables | No | | No | | Yes | | Yes | | Yes | | Yes | |
| | | | | | | | | | | | | |
| # Observations | 1029 | | 250 | | 464 | | 464 | | 441 | | 441 | |
| # Firms | 589 | | 107 | | 348 | | 348 | | 329 | | 329 | |
| | | | | | | | | | | | | |
| log-likelihood | -820.9 | | -251.8 | | -992.6 | | -403.3 | | -602.7 | | -366.7 | |

Capital intensity is another commonly used variable in analyzing the propensity of firms to patent. Capital intensity is usually measured using the ratio of property, plant and equipment (*PP&E*) to the number of employees in the company. For many industries, capital intensity is important as it accounts for the use of capital in the discovery of new and patentable ideas. This seems less important in the software industry, which depends less on equipment and more on the creativity of developers. Arguably, the capital invested in the office latte machine that helps keep employees alert and happy may be as important as the speed of the computers in the office.

Although there are good arguments for excluding capital intensity, it is important to check if the above results are robust to the inclusion of capital intensity in the analysis. A bias could arise from excluding capital intensity if it is an important explanatory variable that is correlated with the *Services* variable or *Fraction of Software Sales*. It is not included in previous estimates because data on property, plant and equipment is not provided in the Software 500 data. To check the robustness of the results from the inclusion of capital intensity, we add information on *PP&E* to the current sample from COMPUSTAT data. The COMPUSTAT data includes only larger firms which leaves just 540 observations from the sample using all patents.[34] Table 8B below repeats the analysis of table 6, but includes the capital intensity variable *log(PP&E/Emp)*.

There are no qualitative changes for either the services variable or fraction of software sales in Table 8B. In addition, the estimates show that only one of the four estimates included in table 8B show that capital intensity is statistically significant. Columns (3) and (4) of the above estimates suggest that after controlling for sector fixed effects, capital intensity does not have a statistically significant effect on patenting. Comparing the estimates to those of Hall and Ziedonis, the estimates in table 10 suggest that relative to the semiconductor industry, the software industry has lower returns to capital intensity while returns to R&D intensity per person is greater.

[34] Similar analysis is done using the sample of software patent firms and we obtain similar results.

Table 9: **Propensity to Produce Patents exlcluding Outliers**

| | (1) Poisson | | (2) Negative Binomial | | (3) Poisson | | (4) Negative Binomial | |
|---|---|---|---|---|---|---|---|---|
| Log(R&D/Employee) | 0.288 | (0.87) | 0.777 | (3.26) | 1.042 | (4.83) | 0.922 | (3.73) |
| log(PP&E/Employee) | 0.304 | (1.10) | 0.420 | (1.95) | 0.144 | (0.68) | 0.260 | (1.21) |
| Log(Employee) | 0.991 | (11.07) | 1.021 | (10.07) | 1.041 | (14.41) | 1.061 | (10.40) |
| Services | -0.027 | -(3.76) | -0.034 | -(4.73) | -0.012 | -(1.85) | -0.025 | -(4.01) |
| Fraction Software Sales | -0.005 | -(1.09) | -0.005 | -(0.88) | -0.004 | -(1.53) | -0.005 | -(0.96) |
| Zero R&D | -22.368 | -(14.60) | -26.823 | -(20.51) | -28.488 | -(19.80) | -30.802 | -(20.51) |
| Year 1999 | 0.221 | (1.07) | 0.648 | (1.68) | 0.287 | (2.32) | 0.559 | (2.04) |
| Year 2000 | -0.461 | -(1.91) | -0.077 | -(0.17) | -0.079 | -(0.47) | -0.147 | -(0.53) |
| Constant | -3.009 | -(2.72) | -1.415 | -(1.21) | -2.834 | -(2.75) | -2.274 | -(1.48) |
| Alpha | | | 3.105 | (4.62) | | | 2.222 | (4.84) |
| Sector Fixed Effects | No | | No | | Yes | | Yes | |
| # Observations | 540 | | 540 | | 512 | | 512 | |
| # Firms | 289 | | 289 | | 270 | | 270 | |
| log-likelihood | -1366.732 | | -618.7961 | | -790.6934 | | -562.8168 | |

*An Analysis of Sector Specific Fixed Effects:*

Several of the previous estimates have included sector fixed effects. This section takes a closer look at the fixed effect estimates to explore the heterogeneity in patenting by sector, controlling for all other variables. Table 10 below shows the sector fixed effects for the Poisson and Neg. Binomial estimates from the estimates in Table 4. Table 10 lists the various software sectors in order of the largest to smallest fixed effects in the Poisson estimates. For example, holding all other factors constant, the Vertical Industrial Applications sector has the highest marginal effect on patenting followed by Application Service Providers. The estimates from the Negative Binomial model are also included in order to check the robustness of the rankings to distributional assumptions. The sectors with the highest propensity to patent include Vertical Industrial Applications, Application

Service Providers, Geographic Information Systems, Operating Systems, Publishing/Graphics, Security, Database, and Wireless/Mobile. There are a number of sectors that seem to patent less than the typical sector. Sectors that tend to have fewer software patents include Financial Applications, Enterprise Resource Applications, E-Business Applications, Enterprise Application Integration, Application Development, Content/Document Management, Business Process Management, and Data Warehouse.

Those results are interesting, in that they parallel similar results in Mann & Sager (2005), showing a substantial difference in patenting practices even within the software industry. Given the limitations of our current dataset, however, we can only speculate as to the causes. In some cases (such as security), they might be driven by early pioneer patents (that make patenting by later entrants more important). In others, they might be driven by the nature of innovation in the particular sector – with higher rates of patenting in sectors in which the innovation is more readily appropriated by patent and lower rates in those in which it is less useful to patent. For present purposes, however, we present the results only for informational purposes.

Table 10: **Sector Fixed Effect Analysis**

| | Software Sector | # Firms | # Obs | Poisson Model | | | Neg. Bino | |
|---|---|---|---|---|---|---|---|---|
| | | | | Fixed Effect | S.E. | Rank | Fixed Effect | S.E. |
| VA | Vertical Indus. Appl. | 15 | 20 | 1.925 | (0.82) | 1 | 0.841 | (0.85 |
| ASP | Application Serv. Prov. | 4 | 4 | 1.883 | (0.45) | 2 | 1.464 | (0.68 |
| GIS | Geogr. Info. Systems | 5 | 7 | 1.607 | (0.69) | 3 | 1.288 | (0.81 |
| OS | Operating Systems | 10 | 14 | 1.591 | (0.75) | 4 | 1.071 | (0.65 |
| PU | Publishing/Graphics | 5 | 5 | 1.473 | (0.58) | 5 | 1.435 | (0.79 |
| SEC | Security | 15 | 20 | 1.392 | (0.77) | 6 | 1.492 | (0.94 |
| DB | Database | 12 | 16 | 1.215 | (0.42) | 7 | 0.785 | (0.68 |
| WVM | Wireless/Mobile | 3 | 4 | 1.202 | (1.24) | 8 | 1.686 | (1.22 |
| SC | Supply Chain | 38 | 54 | 1.159 | (0.70) | 9 | 0.435 | (0.67 |
| CAD | Computer Ass'd Draft'g | 5 | 7 | 1.115 | (0.74) | 10 | 0.636 | (0.82 |
| CRM | Cust. Rel'nship Mgmt. | 32 | 41 | 1.105 | (0.74) | 11 | 0.651 | (0.69 |
| ITS | IT Sourcing | 9 | 13 | 1.030 | (0.65) | 12 | 0.816 | (0.70 |
| SIS | System Integration Servs. | 10 | 16 | 0.856 | (0.73) | 13 | 1.123 | (0.77 |
| SM | Storage Management | 5 | 7 | 0.747 | (0.72) | 14 | 3.518 | (1.20 |
| MW | Middleware | 16 | 19 | 0.686 | (0.51) | 15 | 0.940 | (0.85 |
| BI | Business Intelligence | 20 | 27 | 0.407 | (0.58) | 16 | 0.665 | (0.73 |
| INF | Infrastructure | 39 | 55 | 0.391 | (0.73) | 17 | 0.623 | (0.64 |
| DW | Data Warehouse | 5 | 7 | 0.378 | (0.61) | 18 | 0.596 | (0.92 |
| BPM | Bus. Process Mgmt. | 3 | 3 | 0.369 | (0.47) | 19 | 0.128 | (0.68 |
| CDM | Content/Doc. Mgmt. | 22 | 28 | 0.055 | (0.95) | 20 | 0.253 | (1.06 |
| AD | Application Dev'l'pm't | 42 | 61 | 0.000 | - | 21 | 0.000 | - |
| EAI | Enterprise Appl. Integr'n | 26 | 36 | -0.671 | (0.84) | 22 | -0.576 | (1.00 |
| EB | E-Business Applications | 31 | 44 | -0.852 | (1.09) | 23 | 0.472 | (0.78 |
| ERP | Enterprise Res. Planning | 44 | 62 | -1.409 | (0.72) | 24 | -0.898 | (0.91 |
| FI | Financial Applications | 29 | 42 | -2.258 | (1.14) | 25 | -1.930 | (1.41 |

## 4. Significance and Conclusion

Our paper contributes to the existing literature in several ways. First, it provides a useful extension of the literature (cited in note 5, supra) showing a substantial variation in the use of patents in different industries. Our work provides a much more detailed example of ways in which the use of patents can differ markedly even within the confines of a single industry. Similarly, by providing a quantitative link between patenting propensity and business models, our work provides substantial evidence that patenting, at least in this industry, is a regularized and important part of a well-organized operation, rather than a random or happenstance occurrence. Finally, the paper is important simply for shedding light on the operations of patents in an industry in which they are highly controversial. Although we cannot answer the ultimate welfare question – would the industry be better without patents than it is with patents – we do shed a great deal of light on the reasons why so many firms do – and do not – choose to expend the time and resources necessary to obtain patents to protect their innovative work.