



January 1996

## Book Review

Dennis A. Nicholls

Follow this and additional works at: <http://digitalcommons.law.scu.edu/chtlj>



Part of the [Law Commons](#)

### Recommended Citation

Dennis A. Nicholls, *Book Review*, 12 SANTA CLARA HIGH TECH. L.J. 219 (1996).

Available at: <http://digitalcommons.law.scu.edu/chtlj/vol12/iss1/8>

This Book Review is brought to you for free and open access by the Journals at Santa Clara Law Digital Commons. It has been accepted for inclusion in Santa Clara High Technology Law Journal by an authorized administrator of Santa Clara Law Digital Commons. For more information, please contact [sculawlibrarian@gmail.com](mailto:sculawlibrarian@gmail.com).

---

## BOOK REVIEW

---

SOFTWARE PATENTS. By Gregory A. Stobbs. Wiley Law Publications, 1995. 623 pages.\*

Dennis A. Nicholls†

### INTRODUCTION

Computer software is a troublesome subject matter for those involved in the protection of intellectual property. It is imperfectly protected by copyright and in the past was considered improper subject matter for patent protection. The practice of patenting software has only recently become widespread, due to the Supreme Court's 1981 holding in *Diamond v. Diehr*.<sup>1</sup> *Diehr* paved the way for software patents by correcting the erroneous perception that prior cases had created a *per se* rule that software was unpatentable subject matter.

*Software Patents* is one of the first books to deal extensively with the topic of patenting software. The Patent Resources Group published a book in 1969<sup>2</sup> predating *Diehr*. It appears that the only other recent book on the subject is the volume by Keplinger and Laurie published in 1989,<sup>3</sup> which predates *In re Alappat*.<sup>4</sup> Both of these books are out of date, not only because of the changes in case law, but because of the rapid changes in software technology itself. Stobbs' new book is valuable because of its discussion and practical advice on this controversial subject. However, the book has some shortcomings along with its strengths. The following sections in this review discuss the content and structure of the book. They also make suggestions to allow various readers to make the best use of this first edition.

---

\* Copyright © 1995 by Dennis Nicholls.

† B.A. Physics, University of California, Santa Cruz, 1975; M.A. Applied Mathematics, University of California, San Diego, 1977; Candidate, J.D., Santa Clara University, 1996.

1. 450 U.S. 175 (1981).

2. DAVID BENDER ET AL., SOFTWARE PROTECTION BY TRADE SECRET, CONTRACT, AND PATENT: LAW PRACTICE AND FORMS (1969).

3. MICHAEL S. KEPLINGER & RONALD S. LAURIE, PATENT PROTECTION FOR COMPUTER SOFTWARE: THE NEW SAFEGUARD (1989).

4. 33 F.3d 1526 (Fed. Cir. 1994). *Alappat* held that inventions containing computer programs did not fall under the mathematical subject matter exception unless the claimed subject matter as a whole is a disembodied mathematical concept. *Id.* at 1544.

## DETAILED CONTENTS

*Software Patents* is organized into eleven chapters, each discussing a major topic. The first three chapters provide general background information for the later discussions of current software patents. Chapter one gives a history of patents in general and software patents in particular, with discussions covering the historical period from circa 300 B.C. to the present. Chapter two is a primer on software technology for the practicing attorney; it assumes no prior knowledge of software engineering. This is followed in chapter three by the logical converse: a primer on the legal system for the practicing software engineer. This chapter is very basic, discussing the difference between statutes and case law. The first and third chapters would be interesting to a general reader, but probably not to a patent attorney.

Chapter four is the first chapter to contain substantive practice advice: how to search the prior art. Before prosecuting a patent application, the patent attorney should search the prior art in order to draft the broadest possible claims over what has been done before. The author points out the chicken and egg dilemma of the software patent business. Very few software patents were issued because there was no organized prior art to search, and there was no organized prior art to search because very few software patents were issued.

Stobbs then explains how software patents are classified by the Patent and Trademark Office and why this classification scheme does not serve software well. To oversimplify, inventions are classified by the *result* of the invention, not *how* the result is obtained. The purpose of most software is to provide the means to achieve an end; software does not produce a result *per se*. Consequently, software patents are scattered throughout the prior art. The outcome is that the traditional "shoe search"<sup>5</sup> of a class and subclass is inappropriate for software.

Stobbs proceeds to describe the use of electronic databases to search for prior art references. He covers the use of the Classification And Search Support Information System (CASSIS) CD-ROM tools from the PTO, and the LEXIS and DIALOG on-line services. The author gives a good introduction to using CASSIS. He also points out the importance of using the admittedly limited-scope CASSIS disks:

---

5. According to folklore, the first U.S. patent examiner, Thomas Jefferson, filed copies of patents by classification into "shoeboxes." Searching the prior art by thumbing through a box of copies of similarly classified patents is to this day called a "shoe search" by the examiners in the United States Patent & Trademark Office (PTO). GREGORY A. STOBBS, *SOFTWARE PATENTS* 110 (1995).

the practicing attorney can refine Boolean<sup>6</sup> search strings cheaply on CASSIS before logging onto the much more expensive LEXIS and DIALOG access systems.

Stobbs does not make clear the important distinction between a "database" and a "database access supplier." One does not search LEXIS for patents. Rather, one uses LEXIS, a database access supplier, to search the LEXPAT database. LEXPAT is a library within LEXIS that contains many files. This is an especially significant distinction when using the DIALOG access supplier, for the DIALOG service gives you access, one at a time, to literally hundreds of different databases, which are usually maintained by third parties.

When Stobbs describes searching on DIALOG, he is describing how to access the CLAIMS/U.S. Patents Abstracts database. However, he never mentions the database by name. This is critical because CLAIMS/U.S. Patents Abstracts should be considered superseded by the more comprehensive U.S. Patents Fulltext<sup>7</sup> for the purpose of searching for prior art references. He also confuses the reader about the origin, contents, and utility of INPADOC/Family and Legal Status<sup>8</sup> and Derwent World Patent Index.<sup>9</sup>

Chapters five, six, and seven comprise the centerpiece of the work. They are directed at drafting specifications, drawings, and claims, respectively. Each of these chapters is logically laid out by first listing the requirements of the statute (Patent Act), which is then followed by the regulations (Code of Federal Regulations), administrative language (Manual of Patent Examining Procedure), and the judicial interpretations found in case law. After ensuring that the reader knows what is required in each topic area, the author proceeds in each

---

6. Logical search criteria, with key words connected with logical operators such as "and" and "or," are named after the mathematician George Boole. HANS W. GSCHWIND & EDWARD J. McCLUSKEY, *DESIGN OF DIGITAL COMPUTERS* 24 (2d ed. 1975).

7. LEXPAT and U.S. Patents Fulltext are exact copies of the official, original US PTO database called USPAT, updated each Tuesday morning by the PTO. USPAT contains the entire text of all issued US patents since the early 1970's. The PTO sells copies of USPAT to many companies. USPAT is searched by the PTO's examiners themselves using the Automated Patent System (APS).

8. Stobbs states that DIALOG maintains a database called INPADOC, which he posits is useful for prior art searching. In reality, the International Patent Document Center (INPADOC) in Vienna, Austria, is an organization supported by the World Intellectual Property Organization (WIPO). Their database, called INPADOC/Family and Legal Status, is not very useful for prior art searching for it contains no information (other than the title) describing the subject matter of the patent.

9. The British company Derwent's WPI contains searchable English language abstracts for all included worldwide patents and published patent applications, but searchers are on their own with respect to getting a copy of the patent and having it translated into English.

chapter to give practical advice for one drafting a patent application. All three of these chapters are clear, well written, and highly useful.

Chapter five begins with a review of the textual requirements for a patent specification. Some of this is a general review, although there is some pertinent information on what constitutes undue experimentation with regard to the enablement requirement.<sup>10</sup> Here the author discusses cases holding that the time spent debugging a program should not count towards the time considered in evaluating whether undue experimentation would be required. Then the chapter discusses how to draft a specification. Here the text becomes very specific. Enclosed within this chapter is a checklist for interviewing the inventor, details about what will be covered by the attorney-client privilege, and whether or not to include the source code. The chapter concludes with a section describing the different data structures used by programmers.

Chapter six begins by reviewing the requirement to include drawings in a patent specification. The exposition and detailed warnings (*e.g.*, if the numbering scheme in your specification does not exactly match the numbers appearing on the drawings, you will not get a filing date assigned prior to an office action) lead into a useful discussion on how to convert design documentation into drawings for the patent application. No software is written without large numbers of design and development drawings being generated and the author shows how to recycle these drawings into the patent application. The kinds of drawings discussed include old fashioned flowcharts, Grady Booch's notation for object oriented programming,<sup>11</sup> and Yourdon diagrams for system design.<sup>12</sup>

The final chapter in the central trilogy, chapter seven, deals with claim drafting. Surprisingly, this chapter does not spend much time on the issue of patentable subject matter. Stobbs leaves this to a subsequent chapter. One needs to know what subject matter will be allowed prior to drafting a claim. Also, it is common practice in patent application drafting to draft the claims first and the rest of the specification later. The author repeats this advice, although in the previous chapter advised to do the drawings first to help the drafter determine

---

10. The specification must enable a person skilled in the relevant art to make and use the invention. 15 U.S.C. § 112 (1988). Some experimentation is allowable, but there must not be undue experimentation before the invention can be made and used. *W. L. Gore & Associates v. Garlock*, 721 F.2d 1540, 1557 (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984) (citing *In re Angstadt*, 537 F.2d 498, 503-504 (C.C.P.A. 1976)).

11. Booch notation uses four views of a software model, showing classes, objects, modules, and processes. STOBBS, *supra* note 5, at 209-10.

12. Yourdon/DeMarco data flow diagrams use arrows to show the path over which data are passed. These diagrams may be built upon an existing software system, and thus are most useful in showing an advance over the prior art. *Id.* at 215-20.

the actual inventive step. On the whole this chapter goes into great and helpful detail, showing how to draft a claim to avoid the standard rejections during prosecution.

Following a short chapter eight on the mechanics of the patent office's processing of an application, in chapter nine the author tackles the most central and controversial part of the software patent business: the allowable subject matter under 35 U.S.C. § 101 of the Patent Act. Recall that until recently software was simply not considered patentable subject matter. This chapter discusses the cases leading up to, and including, *Diamond v. Chakrabarty*,<sup>13</sup> *Diamond v. Diehr*,<sup>14</sup> and *In re Alappat*<sup>15</sup> and synthesizes their holdings. A section containing case digests then follows, which shows what kinds of software subject matter have been found patentable under § 101 and what kinds have not. This allows the practitioner to forecast whether or not his software is patentable. The chapter ends with Stobbs' own rule for finding patentable subject matter: the human control test. As Stobbs observes:

[W]hen confronted with the dilemma of patentable invention or unpatentable principle, ask yourself about the claimed invention, "Can humanity control this thing? Does humanity have the power to change this thing, or make this thing cease to be true or cease to exist?" If so, then this thing, if new, may be patented, for it qualifies as "anything under the sun that is made by man."<sup>16</sup>

After a digression into European Patent Office (EPO) practice in chapter ten, the author ends his book in chapter eleven with a discussion of several dozen illustrative software patents. These are conveniently listed by assignee. Not surprisingly, software giant Microsoft is the assignee in twenty-seven of these patents. These patents not only serve as a paradigm of claims drafting but may also function as an introduction to the prior art.

#### TARGET AUDIENCE

Book reviewers often begin their reviews with a comment on the anticipated audience that the author is trying to reach and what they should gain from the book. After reading the book, this reviewer is uncertain about who constitutes the target audience. Unfortunately, the author himself gives no clue in his preface. It is unclear whether the book is intended to serve as a textbook to be read right through as the centerpiece of a short course on software patents, or if the book is

---

13. 447 U.S. 303 (1980).

14. 450 U.S. 175 (1981).

15. 33 F.3d 1526 (Fed. Cir. 1994).

16. STOBBS, *supra* note 5, at 328-329.

intended to serve as a hornbook for random reference. Perhaps the author's intent is to have the patent attorney loan the book to a client as part of the counseling process.

If the book is intended to serve as a text for a short course on software patents, then the reviewer would suggest that the reader study the chapters in the order 9, 4, 2, 6, 5, 7, 11, and 8. This will help the reader to focus on the appropriate subject matter for software patents before learning about prior art searches and drafting patent applications.

If the book is intended to serve as a hornbook or treatise, then the ordering of the chapters is less important and the need for somewhat more comprehensive cross-referencing becomes apparent. Chapters two and six contain much overlapping information about software design and should be read together. Chapter three is too elementary for legal practitioners. Chapter nine's comparison of the subject matter allowable under U.S. law to the plain language of the EPO statute leaves one with the mistaken impression that software patents are all but impossible in the EPO.<sup>17</sup> Here, a simple reference to chapter ten, section four, would direct the casual reader to the interpretation of the European statute by European courts that allowed software patents in the EPO.

#### CONCLUSION

*Software Patents* by Gregory A. Stobbs contains a large amount of information useful to the patent attorney seeking to prepare and prosecute an application for a software patent. The chapters on drafting the application are very clear and helpful. Unfortunately there are also certain shortcomings in the format and content of the book, for a current book about obtaining patent protection for software is very much needed.

The chapter on searching the prior art is the most problematic. It was not clear to the reviewer if the chapter was intended to be encyclopedic or merely give a few unconnected helpful tips on searching the prior art. If the intent was to be encyclopedic, then too many methods of searching were omitted. The use of DIALOG was discussed, but not the similar access supplier Questel-Orbit.<sup>18</sup> Access via the Internet was discussed, but perhaps the single most important re-

---

17. "The following in particular shall not be regarded as inventions . . . programs for computers." Convention on the Grant of European Patents, Oct. 5, 1973, art. 52(2), 13 I.L.M. 270, 285, reprinted in STOBBS, *supra* note 5, at 277.

18. Questel-Orbit is an on-line service offered by France Telecom. Many of the databases searchable under DIALOG are also searchable under Questel-Orbit.

source on the Internet was missed — Electronic Data Services' *Shadow Patent Office*.<sup>19</sup> If, on the other hand, the chapter on searching the prior art was meant to give a few helpful tips, this more limited purpose should be made clear. Furthermore, the potential for confusion caused by some of these tips, mentioned above, should be removed.

The reader should also be made aware that much of the book is of general, rather than specific, interest in the field of software patents. The introductory chapter on the history of patents is interesting, and at times amusing. However, it is of limited use to the patent attorney trying to prosecute a patent application. Similarly, the enormous Appendix A<sup>20</sup> contains a verbatim transcript of PTO public hearings on the use of the patent system to protect software related inventions. However, it is of limited use in its unedited form.

---

19. Shadow Patent Office is a service that allows the user to enter a textual description of the invention and receive back a listing of closely related patents. It may be reached at the following addresses: on the Internet at [spo\\_patent@spo.eds.com](mailto:spo_patent@spo.eds.com), and on the World Wide Web at <http://www.spo.eds.com>.

20. Appendix A takes up 175 pages out of a total of 623 for the book.

