

Unless otherwise noted, the content of this course material is licensed under a Creative Commons Attribution – Non-commercial – Share Alike 3.0 License.

<http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Copyright © 2008, Dale Myers.

You assume all responsibility for use and potential liability associated with any use of the material. Material contains copyrighted content, used in accordance with U.S. law. Copyright holders of content included in this material should contact open.michigan@umich.edu with any questions, corrections, or clarifications regarding the use of content. The Regents of the University of Michigan do not license the use of third party content posted to this site unless such a license is specifically granted in connection with particular content. Users of content are responsible for their compliance with applicable law. Mention of specific products in this material solely represents the opinion of the speaker and does not represent an endorsement by the University of Michigan. For more information about how to cite these materials visit <http://michigan.educommons.net/about/terms-of-use>.

Any medical information in this material is intended to inform and educate and is not a tool for self-diagnosis or a replacement for medical evaluation, advice, diagnosis or treatment by a healthcare professional. You should speak to your physician or make an appointment to be seen if you have questions or concerns about this information or your medical condition. Viewer discretion is advised: Material may contain medical images that may be disturbing to some viewers.

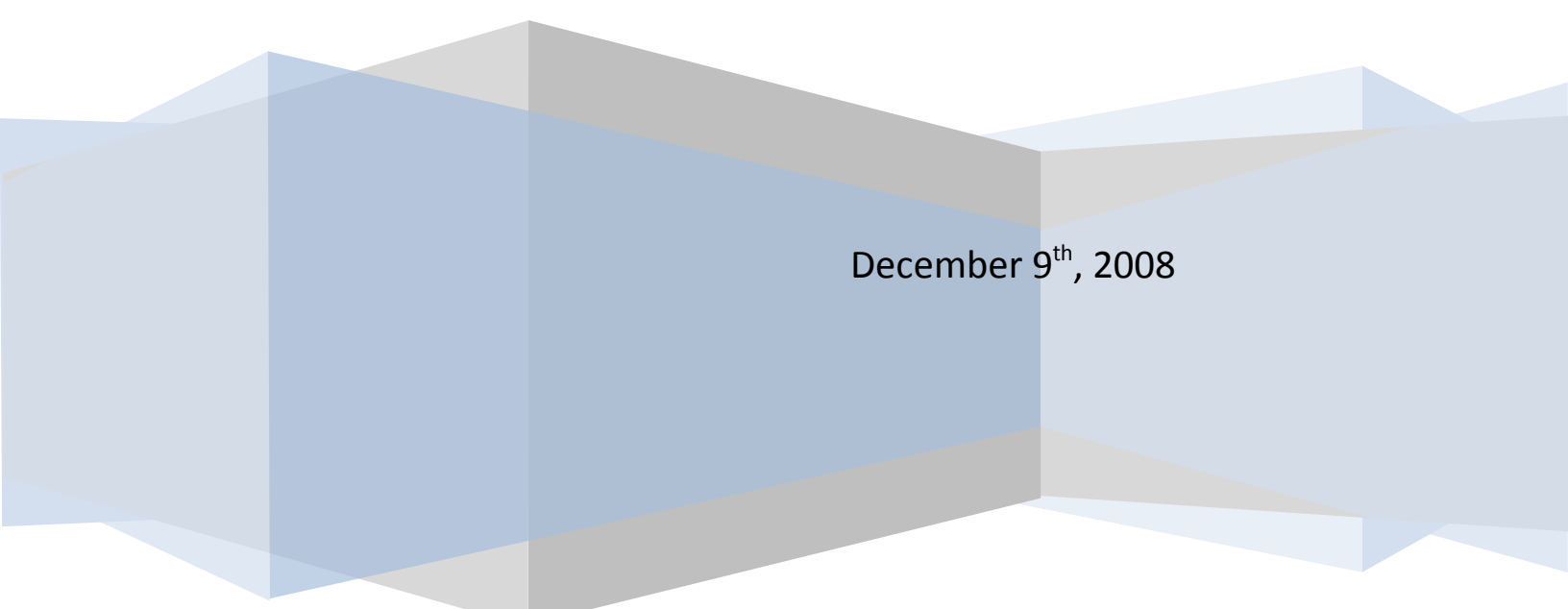
“SI 519”

No Rest for the Weary

**Changing Patterns of Software, Databases and
Their Intellectual Property Protection**

Dale Myers

December 9th, 2008



Introduction

Since the early 1980's there have been various types of debate over the intellectual property protection for software and databases including copyright, patent and trade secret protection. The debate is always focused on providing the best legal protection while encouraging innovation and progress. Copyright arguments have involved seemingly different interpretations of idea expression dichotomy. Debate in the patent context has considered whether some software patents have actually discouraged rather than encouraged innovation. Some have even argued that the need for patent protection is questionable and that it is reasonable to rely on copyright and licensing to encourage software innovation.¹ Trade secret protection has been called into question when federal copyright precludes its application in specific legal cases.

As a result of these debate and resulting questions, patent had become the preferred method for seeking protection of software and copyright for databases. However, the recent Bilski ruling may dramatically change the landscape for software patent protection creating even more debate and questions.² One thing is clear. Intellectual property laws and those who write them need to more frequently examine existing IP protection and how it applies or doesn't, to databases and software. The technology and the industry have changed radically in the last 10 years, but the laws have not.

It is worthwhile to look at the current laws, the issues they can create and potential solutions. When the structure of an industry changes dramatically and the regulations are not examined to determine whether or not they are still effective, the risk of inadvertent overregulation and poorly applied legal principles increases exponentially. The assurance of freedom of expression and incentive to innovate provides compelling reasons for this examination.

¹ Simson L. Garfinkel, Richard M. Stallman & Mitchell Kapur, *Why Patents are Bad for Software*, ISSUES IN SCI. AND TECH., Fall 1991, at 52-55; Eloise Gratton, *Should Patent Protection be Considered for Computer-Software Related Inventions*

² <http://www.patentlyo.com/patent/2008/02/bilski-full-caf.html>

Current Intellectual Property Law Protection of Software

To frame the discussion of the current IP software protection mechanisms, we should first have a good definition of software. In the software trade there are two types;

- Systems software which includes operating systems and utilities which enable a computer to function
- Application software which includes programs which create results from a computer for an end user. Application software includes word processors, spreadsheets and database management systems.

Software can be protected through copyright, patent or trade secret which comes from contract law.

Examining each form of protection's relevance to software facilitates a later assessment of the effectiveness of each one.

Copyright

Copyright law considers software to be a "literary work".³ Just like the owner of a copyright for a book, the copyright owner of software has exclusive rights to make and distribute copies and create derivative works. The software owner is the developer, or author, who automatically acquires the copyright as soon as it is "fixed in a tangible medium",⁴ in this case perhaps written on paper, but most likely entered into a computer in readable form. The copyright lasts for the life of the author plus 50 years if the author is a person, or 150 years if the author is a corporation. The authors of software may sell or license each or all of these rights. Over two decades ago Congress and the National Commission on New Technological Uses of Copyrighted Works (CONTU) agreed that copyright protection constituted the best

³ <http://www.copyright.gov/title17/92chap1.html#102>

⁴ <http://www.copyright.gov/title17/92chap1.html#102>

available option for protecting the rights of software owners. Their reasoning was instinctive, but also based on the fact that while primarily functional, computer programs contained symbols, terms and numbers that are normally associated with literary works.⁵ As a result, Congress only made minor amendments to the Copyright Act for software to include the definition of a “computer program” and the rights of owners to make copies for archival purposes or in the course of using them for their intended purpose.⁶

Patent

Software can also be protected under patent law. It must meet the requirements for obtaining a patent in that it must be statutory, new, useful and non-obvious.⁷ Software developers and owners often seek patent protection since it can provide stronger protection than copyright law. This is primarily due to the fact that copyright protection only applies to the expression of an idea and not the idea itself. Software owners have also gravitated to patent protection due to its broadness. To date, they have been attracted by the US Congress’s prior statement that “anything under the sun made by man” deserves patent protection to promote innovation.⁸

The notion that patents can promote more innovation in software has also led to increased use of patents for protection. Patents’ ability to provide a limited time monopoly for new software increases its value and potential return to the owner. Additionally, since the law requires that patent owners disclose their inventions publicly, there is an expectation that additional development will be stimulated and that new demand can be met.

⁵ Copyright Law and Subject Matter Specificity: The Case of Computer Software, Stacey L. Dogan and Joseph P. Liu, [http://www1.law.nyu.edu/pubs/annualsurvey/documents/61%20N.Y.U.%20Ann.%20Surv.%20Am.%20L.%20203%20\(2005\).pdf](http://www1.law.nyu.edu/pubs/annualsurvey/documents/61%20N.Y.U.%20Ann.%20Surv.%20Am.%20L.%20203%20(2005).pdf)

⁶ 17 U.S.C., Sections 101,117

⁷ <http://www.bitlaw.com/patent/requirements.html>

⁸ *Diamond v Chakrabaty*, 447 U.S. 303, (1980)

Trade Secret

Trade secret protection can also be applied to software in its development phase and to basic software instructions known as source code. Trade secret protection is more commonly applicable when a company develops software and wishes to keep it and its functions private. Trade secret protection also involves a contract, typically between the developer and the owner. A common example is one where contract developers are hired by a firm to develop software. Those contractors are typically asked to sign a contract or non-disclosure agreement that assigns IP rights to the firm and agrees to keep the development a secret.

Current IP Law Protection Databases

The same intellectual property protection mechanisms apply to databases. Again, it is important to understand the definition of a database;

- In the computing industry a database is a collection of information that computer programs action to create function and present information to the end user
- In legal terms, US Copyright Law defines a database as a “compilation: a work formed by the collection and assembling of preexisting materials or of data.” It also defines an automated database as “a body of facts, data, or other information assembled into an organized format suitable for use in a computer and comprising one or more files.”⁹

Copyright

Given copyright law’s definition of a database, it is protected as a compilation provided that the data is selected, coordinated and arranged in such a way that qualifies the database as an original work of

⁹ <http://www.copyright.gov/title17/>

authorship. In that case, the database is protected in much the same way and with the same owner rights as software. A slight difference is introduced with regard to derivative works. Copyright only extends to the material the author has actually contributed to the work and not to any pre-existing materials she may have used to populate the database. The seminal case that enables understanding of how the law views copyright protection for databases is *Feist Publications Inc. v. Rural Telephone Service Company*, 499 U.S. 340 (1991). In this case the U.S. Supreme Court qualified and quantified necessary elements of a database in order for it to be protected by copyright. Facts, the court held are not eligible for copyright protection. However, if those facts are a compilation and if the author has selected and arranged them in an original and minimally creative way, the work can be protected under copyright law. The court also cited limitations for protection relating to the protection of the actual elements of the compilation, requirements the author must meet for copyright to apply and a definition of expression, which is essential to copyright protection. The court also admitted that copyright protection for a factual compilation is “thin,”¹⁰ something which will be examined in greater detail later.

Patent

It is unlikely that patent protection could ever apply to databases. By their nature, they are collections of work that has already been published and do not meet the statutory, new and non-obvious requirements an owner needs to obtain a patent.

Trade Secrets

Trade secret protection does apply to databases with many of the same conditions as apply to software. The critical condition is the presence of a contract or non-disclosure agreement between the owner and

¹⁰ *Feist Publications Inc. v. Rural Telephone, Service Company*, 499 U.S. 340 (1991)

another party. Further, the Uniform Trade Secret Act provides for injunctive relief and damages when people “misappropriate” a trade secret.¹¹ Further clarification on this protection and on factual compilations in general came from the 2nd Circuit in *NBA v. Motorola*, 105 F.3d 841 (2nd Cir. 1997). In this case the 2nd Circuit overturned the lower court, ruling that federal copyright prohibits the ownership of factual material and therefore that state law regarding trade secrets could not be applied to the databases in this case. However, they did further clarify what constitutes misappropriation and how trade secret protection can apply to databases.

Issues with Current IP Protection

Significant legal and software industry analysis of how intellectual property protection mechanisms apply to software and databases has occurred in recent history, which is to be expected as any new media is first introduced and then evolves. However, the primary concern in this discussion is how courts have chosen to apply and interpret the laws and the downstream, perhaps unanticipated implications of those interpretations. A recurring theme in infringement cases is that the courts tend to take a case by case approach. As a result, software and database developers have expressed a great deal of confusion and apprehension regarding how the courts will rule in specific cases.¹² This phenomenon has occurred because the courts have been faced with the unenviable tasks of balancing the the rights of a free society with maintaining individual incentive to innovate. Understanding these issues in more detail will facilitate investigation of new options for software and database IP protection the courts or Congress could consider.

¹¹ <http://nsi.org/Library/Espionage/usta.htm>

¹² <http://www.fcsuper.com/swblog/?p=118>

Software

The primary issues with software are when copyright and patent are invoked. The most compelling of these issues are;

- Confusion regarding what is protected by copyright. This confusion has been caused primarily by the idea/ expression dichotomy of copyright protection and its application to software.
- Increased applications for and awarding of patents to protect software due to insecurity regarding copyright protection.
- Significant increases in software development costs associated with acquiring patent protection and its negative effects on software development cycle.

Confusion

One can argue that the interpretation of copyright law can and has been inconsistent leading to the contention that there is much confusion in the software development community regarding copyright protection. This can be seen by comparing rulings in *Whelan Associates v Jaslow Dental Laboratory* (1986) and *Computer Associates International Inc. v Altai Inc.* (1992).¹³ In *Whelan*, the Third Circuit Court applied the freedom of expression rule to a computer program in this manner; “The ‘expression of the idea’ in a software computer program is the manner in which the program operates, controls and regulates the computer in receiving, assembling, calculating, retaining, correlating, and producing information either on a screen, print-out or by audio communication.”¹⁴ Even though the program in question in this case was not original in its function, performance, or coding, the court ruled that it was

¹³ <http://otd.harvard.edu/inventions/ip/software/compare/>

original and deserving of protection due to its “original structure and organization.” In the Computer Associates case the Second Circuit Court applied an “abstraction-filtration comparison” test that established that non-original elements of a software program had to be removed before it can be copyrighted. With regard to Whelan they wrote, “We think that Whelan’s approach to separating idea from expression in computer programs relies too heavily on metaphysical distinctions and does not place enough emphasis on practical considerations.”¹⁵ While these rulings are not seemingly contradictory from a legal standpoint, they are counterintuitive to software developers, considering that the program in Whelan had essentially been copied with the only changes being to make it run more efficiently. The program in the Computer Associates case was a perfunctory job scheduler reusing standard interfaces to the MVS operating system, even though Altai claimed it had been developed in a “clean room,” meaning Altai developers had not seen CA code. While in both cases the courts ruled that the programs were protected, the rationale is confusing. It is therefore not surprising that individuals and corporations have perceived patent protection as clearer and more reliable.

Patent Spree

In 1995 8,571 software patents were granted out of a total of 101,149 granted by the USPTO. In 2007, 38,874 software patents were issued out of 185,710.¹⁶ As mentioned previously, there is widely held opinion that this explosive growth came from confusion about and lack of confidence in copyright protection for software. In an article published in 2004, Microsoft’s General Counsel and another of their lawyers attributed the growth of the software industry to its IP protection. They went a step further, “without IP protection, second-comers could simply copy the innovation and thereby appropriate at least some portion of its economic value, without having to bear any related development costs. The possibility that third parties might “free ride” on the original inventor’s investment in this

¹⁵ <http://www.bitlaw.com/source/cases/copyright/altai.html>

¹⁶ http://commons.wikimedia.org/wiki/Image:US_granted_software_patents.png

manner increases the risk that the developer might be unable to earn a competitive return on this investment in the marketplace, thereby diminishing or even eliminating the inventor's incentive to invest in future innovations.”¹⁷

This type of growth and opinion in any activity may be legitimate, but also likely drives a dramatic increase in frivolous activity. As a result, many in the software industry, supported by concerned legal resources have called for patent reform as it relates to software. These most common calls for reform are centered on the following premises;

- Third-party participation in the patent examination process;
- Post-grant oppositions
- Challenges to the validity of patents to be based on a "preponderance of the evidence," not "clear and convincing evidence" standard;
- Restoration of the once-strong non-obviousness standard;
- A higher threshold in finding willful infringement;
- A higher threshold before granting injunctive relief;
- Publication of all patent applications after 18 months;

- Harmonization with Europe to narrow the scope of what is patentable in the software and business method arenas¹⁸

Though no new legislation has been passed recently regarding patents, clearly these issues bear continued examination as Congress has considered patent reform in 2003, 2005 and again in 2007. It would be remiss not to mention that meaningful changes could occur with regard to software patents in light of the recent *Bilski* decision. In this decision the court applied a two pronged qualification of software being “tied to a particular machine,” and possessing the capability to transform an “article” into a different state or thing, only future rulings will reveal how courts will apply these qualifiers.¹⁹

¹⁷ Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents, 71 U. Chi. L. Rev. 241 (2004).

¹⁸ <http://www.law.duke.edu/journals/dltr/articles/2005dltr0012.html>

¹⁹

<http://www.mondaq.com/article.asp?articleid=69582>

Software Patents and the Development Cycle

Many in the software industry have argued that the propensity for companies and developers to seek patent protection has had an extremely negative effect on productivity in the industry. Opponents of over patenting of software point to two main factors. First, they contend that there is prohibitive cost in the research and application activity necessary to obtain a patent. They also express the concern that the research one needs to obtain a patent can never be thoroughly completed and that developers may already be paying license fees on software protected by patent. Second, opponents point to the amount of development and deployment delays that occur while waiting for patent approval. According to the USPTO web site, the average amount of time for a pending patent was 3 and one half years, which is a lifetime in the software industry. The opportunity costs of this type of delay for businesses, who are trying to gain competitive advantage, make patent protection untenable. Combine this with the fact that many patent applications are made public after 18 months, the patent fails to achieve the protection many corporate and individual developers are seeking.

Databases and Copyright

As stated earlier, copyright is the primary IP protection mechanism for databases. The issues with the protection can best be illustrated by examining a small database and its elements. Consider the following small database;

<i>Title</i>	<i>Department</i>	<i>Major Subject</i>	<i>Minor Category</i>	<i>Quote</i>	<i>Comment</i>
<i>Gone with the Wind</i>	<i>Cloth</i>	<i>Fiction</i>	<i>Civil War</i>	<i>"I don't know nuthin' 'bout birthin' babies"</i>	<i>Mitchell epic of antebellum south</i>
<i>Fargo</i>	<i>Video</i>	<i>Comedy</i>	<i>Thriller</i>	<i>"We're not a bank Jerry."</i>	<i>Coen brothers best effort</i>
<i>A Tale of Two Cities</i>	<i>Trade Paper</i>	<i>Fiction</i>	<i>Classics</i>	<i>"It was the best of times"</i>	<i>Not the Broadway musical</i>
<i>The Godfather</i>	<i>Video</i>	<i>Crime</i>	<i>Mafia</i>	<i>"either his brains or his signature would be on the contract"</i>	<i>The greatest movie ever made</i>

Several key questions must be answered to determine whether or not the database or only certain elements of the database are copyright protected. They are;

- Does the selection and arrangement of the elements qualify the collection to be copyright protected as an "original work of authorship?"
- What is the impact of having added elements that are potentially not pre-existing?
- Are the comments original works and do they make the entire compilation protectable?
- Are there other copyright limitations for some elements that might render the database unprotected?

As noted earlier, in *Feist Publications Inc. v. Rural Telephone Service Company*, 499 U.S. 340 (1991), the Supreme Court set down qualifiers for what is and is not protected by copyright in the case of a compilation. The Court stated that facts are not copyrightable, but that if some originality is applied in

their “selection and arrangement” and the compiler has acted independently, that “Congress may protect such compilations through the copyright laws.” Can that logic be applied to the sample database? Certainly the titles of the books and movies are facts and therefore not protected, or are they? The title “The Godfather” could certainly still be protected by copyright as is “Gone with the Wind.” In Feist, the court also laid out requirements about “clothing” facts to qualify the compilation for protection. In this case the author has attempted to do that by adding descriptors such as “Department” and “Major Subject.” Do these descriptors provide enough clothing for protection? An argument could be made that they are public and common descriptors and therefore do not qualify as clothing unlike the comment field of the compilation. Clearly the Court recognized the issues as their key qualifier in the Feist ruling was that, “the copyright in a factual compilation is thin.”

Once can certainly argue that the thinness factor creates some of the same ambiguity for database protection as we examined earlier for software. In “Kregos v. Associated Press, (40) the Second Circuit found the plaintiff’s “pitching form” — a form comprised of nine statistics about a baseball pitcher’s performance — copyrightable.”²⁰ The court believed that selection of those 9 statistics from all that could have been selected constituted “requisite originality.” Conversely, in Victor Lalli Enterprises, Inc. v. Big Red Apple, Inc.,(42), this same court found that a selection of lottery numbers arranged in a grid was not original because the numbers were computer generated.²¹ The degree of “thinness” in these two cases could certainly be open to debate. In Victor Lalli, the numbers were indeed originally selected, just by a computer. As a result, database producers have become more creative in attempts to make databases copyrightable. A popular tactic is to add material known to be protected under copyright. The problem here is that this data is often not relevant or relational to the rest of the database, which creates unnecessary computing overhead. Other producers have attempted to make their databases

²⁰ U.S. COPYRIGHT OFFICER REPORT ON LEGAL PROTECTION FOR DATABASES,, August 1997

²¹ U.S. COPYRIGHT OFFICER REPORT ON LEGAL PROTECTION FOR DATABASES,, August 1997

more creative by applying very specialized arrangement criteria. The issue here again is that there are technical standards for the methods of arranging data to achieve optimal computing performance. Circumventing these standards to enhance the possibility of string copyright protection decreases performance and increases costs. Realizing this, many database creators now rely on contracts and trade secrets to protect their data.

Trade Secrets for Software and Databases

State law and contracts are frequently used to protect both software and data. This is mostly the case for corporations who fear the loss of their competitive advantage by exposing their data to the public. However, as in the other two protection mechanisms there are issues;

- Many contracts of this type are executed in “Terms of Use” agreements, which as learned in *Bragg V Linden Labs*²², do not always provide the expected level of protection.
- Federal copyright law can preempt a contract or state trade regulations as in the case of *NBA v. Motorola*, 105 F.3d 841 (2nd Cir. 1997).
- Contracts do not bind third parties who might come into possession of software or a database. Software distributed on a CD, for example, is acquired by a third party who was not party to some initial contract.

Given the issues with copyright, patent and trade secret protection for databases, taking on a challenge to think in a more temporal manner about this media is appropriate.

²² Marc Bragg v Linden Research Inc., and Philip Rosedale

New Possibilities for Intellectual Property Protection for Software and

Databases

While courts have found ways to effectively interpret and apply IP protection (particularly copyright and patent) to software and databases, many in the software industry feel that significant changes are required. Influential leaders in the industry have complained aggressively about the current forms of protection. Andy Grove, former chairman of Intel has stated, "the US needs to revamp not just the patent system, but the entire system of intellectual property law."²³ The controversy rages in many quarters, but there are three prominent and recurring issues that need to be addressed;

- The protection of components of technical systems comprised of software and databases as opposed to an entire system.
- The idea within the technology industry that reuse of existing software and databases is a cornerstone of new development strategies and methods. Published and reused source code and data enables the industry to advance at a rapid pace.
- Current IP statutes' potential restriction of interoperability between systems, which is another concept that is critical to intellectual advancement and economic viability of software and databases. Disparate systems must be able to exchange and use information.

Congress can improve the current intellectual property laws by focusing on these three areas as they provide the best examples of the disconnection between current laws and the direction of the technology/software industry.

²³ <http://www.washingtonpost.com/ac2/wp-dyn/A54548-2003Dec10>

Software and Databases as a Single Entity

Software and databases are inextricably connected. Software cannot produce results without a database and automated databases, though not quite yet, will become irrelevant without software. One can argue that much of the confusion regarding IP protection for both has resulted from the sometimes very narrow approaches that courts have taken particularly when attempting to apply copyright protection. Nonetheless, courts have been consistent in trying to establish “requisite originality”²⁴ requirements when deciding whether or not software and databases are protected by copyright. In *Feist Publications Inc. v. Rural Telephone Service Company*, 499 U.S. 340 (1991) for databases and *Whelan Associates v. Jaslow Dental Laboratory* (1986) for software, nearly the same application of a test for “expression” was applied. It seems logical then to suggest that establishing software and databases as a single system makes sense from an IP protection perspective.

Assuming that expression is a key factor in determining how software and databases are protected, viewing them as a single work could provide stronger protection. Creating or developing a system including software and a database requires much more creativity and expression than simply writing code or selecting and arranging elements of a database. The protection of a composite system would be much stronger, less confusing and potentially assure that the expression of the authors is the focus rather than elements they may have added, which may or may not be protected. These elements are trivial when focusing on the function of a system, which should be the main goal of copyright protection. The practical application of software and databases is to produce a viable result for an end user by acting together. Continuing to legally treat them as separate entities and focusing on elements as opposed to function will weaken the protection and cause even more confusion in the software community about copyright protection.

²⁴ *Feist Publications Inc. v. Rural Telephone Service Company*, 499 U.S. 340 (1991)

Software and Database Reuse Strategies

In the early part of this decade, new software architecture was created. The objectives of this architecture were to speed the development process and to have developers be able to reuse core software to build new function. This architecture relies on software “components” and is known as “Component Based Development.”²⁵ Developers use components, which use pre-determined partitions of databases, to build new software enabling new features and functions. This architecture essentially provided a technical enabler to the open source software community to better share outstanding development achievements that could be freely reused. Development of the component architecture represented the physical change to an industry that was already experiencing radical policy change through open source licensing. Initially, licensing of open source provided the original developer(s) with good control over the software they wanted to share. With components, new development occurred much faster making it a challenge to keep track of how a developer’s open source code and data were being used and if the use complied with the licenses.

While component development of software and databases is by no means an exclusively used method in the industry, it is an excellent representation of an architectural change intended to bring about significant improvement and benefits that could be inhibited by existing intellectual property law. This change in software and database architecture is not unlike the architecture changes Lawrence Lessig discusses regarding the Internet and its regulation.²⁶ By not changing regulatory practice, or in this case not knowing how to modify copyright and patent laws for databases and software, legislators could unintentionally over regulate the industry. Two simple changes to existing patent and copyright protection could mitigate this potential problem for the short term.

²⁵ <http://cbs.colognet.org/overview.php>

²⁶ Code: Version 2.0, Lawrence Lessig, Basic Books, (2006)

- Change the length terms of protection for software and databases to 5-7 years. This would provide original developers with an opportunity to gain from their creativity, yet support the objectives of open source and component development. It will also reduce the amount of research needed when applying for patents or trying to avoid copyright infringement.
- Require that all, not just some as is currently the case, software and database (should they ever exist) patents be published at 18 months. This increases the profile not only for potential reuse (legal of course), but also creates a higher profile for detecting potential infringement.

These changes and more like them would create improved synchronization between the goals of the software industry and those charged to protect intellectual property.

Interoperability

Another significant change in the software and database industry in the last decade is the notion of interoperability. The idea of interoperability emanates from users of software, who may work on disparate systems, but have requirements to have those systems exchange data, file formats and protocols.²⁷ Many systems and programs are designed to run on specific operating systems and chip sets or platforms. This had rendered them incompatible until the developers realized they could write “middleware”, which would enable the systems to interoperate. This middleware often requires copying of code and logic from an original system to enable the systems to interact. In fact, building an interoperable system almost requires infringement when trying to work with patented software implementing a process called reverse engineering. In this process, the developer of the new program breaks down the compiled binary code to discover the source she needs to copy the program with which she wants to interoperate. While in *Bonito Boats v Thundercraft*, the Supreme Court has held that “the competitive reality of reverse engineering may act as a spur to the inventor, creating an incentive to

²⁷ <http://en.wikipedia.org/wiki/Interoperability>

develop inventions that meet the rigorous requirements of patentability,²⁸ for an engineering process, reverse engineering for software is less clear. Anti-reverse engineering licensing and the Digital Millennium Copyright Act's (DMCA) provisions against using anti-circumvention mechanisms in software have complicated this issue. Again, courts are faced with using IP law not written in the interoperability context to protect innovation in the industry. Interoperability issues for software are not going to go away for two main reasons;

1. Developers, fearing assessment of damages when they knowingly use patented software, will be discouraged from innovating in the interoperability context.
2. Unrealized network externalities in the distribution of interoperable software will have extremely negative effects on the software industry economics and the economy in general

One reasonable solution would be to adopt an interoperability exception to software patents. This will require a very narrow and specific definition of interoperability. However, since it is not a completely new concept in the software world that seems possible. In fact, Sun Microsystems has proposed language including words like "indispensable" to qualify the exception for copying or reverse engineering a program.²⁹ Another real possibility stems from the recent *Bilski* decision. Interoperability could be determined to be a condition of transformation in that a process that enables two disparate systems to interact is indeed transformative. Both changes would enable developers to use existing software to create interoperable without fear of infringement promoting innovation and the distribution of software everyone can use.

²⁸ *BONITO BOATS, INC. v. THUNDER CRAFT BOATS, INC*, Supreme Court of the United States, No87-1346

²⁹ Sun Home Page, Position Papers: Software Patents, http://www.sun.com/aboutsun/policy/software_patents.html

Conclusions

Courts have done an effective job of counterbalancing the need for innovation while protecting the freedom of expression pertaining to software and databases. However, in many cases, the applicable rulings and precedents were derived from IP laws applied to compilations rather than automated databases, as in the Feist case, or unrelated technology as in the Bonito case. These laws were not written for and do not take into account the constant change that occurs in the software industry and that lawmakers and interpreters do not always understand those changes or the technology they are regulating.

While Andy Grove's statement regarding IP law was extreme there is clearly a need to examine IP laws and their application to software and databases to effect improvements in protection and promotion of freedom. It is possible to achieve this assuming Congress will revisit Patent Reform and re-examine copyright protection. This is especially true since newly presented IP cases, given the Bilski ruling, present opportunities for judges and lawyers to motivate Congress to examine and reform specific IP statutes. Without these assessments it is very possible that the industry will suffer with very narrow and misapplied existing law.