"The potential power of 'Software Patents' to destroy Crystallographic Software; Crystallographic Software Development; and Crystallography's Future."

(Full article of a summary to be printed in the British Crystallographic Association, December 2002 Newsletter (Crystallography News) - date submitted: 26th September 2002 – minor updates and corrections: 2nd October 2002 - "Advice for Inventors" and comments on "frivolous litigation countersuits" added on 7th October 2002 – further minor updates 8th October 2002. Minor updates from Bruce Perens' articles 26th October 2002. And minor text update on 6th Feb 2003. Text update on 8th Feb on feasibility of challenging European Software Patents)
Available on the web at the BCA via : http://bca.cryst.bbk.ac.uk/bca/cnews/xtlnews.htm
Available on the web at the CCP14 via : http://www.ccp14.ac.uk/maths/software-patents/

by Lachlan M. D. Cranswick
CCP14 for Single Crystal and Powder Diffraction
School of Crystallography, Birkbeck College,
Malet Street, Bloomsbury, WC1E 7HX, London, UK
E-mail: l.m.d.cranswick@dl.ac.uk   WWW: http://www.ccp14.ac.uk/

Current member of the IUCr Commission on Crystallographic Computing
Secretary of the European Crystallographic Association's Computing SIG

Table of Contents

Introduction

While it is possible to write an objective essay on the issue of "Software Patents" (as could be done of "gas chambers"), would you really want to? Especially if more concerned for its victims and "victims to be", in preference over a minority who would profit by their use. This article is concerned about the negative effects and chaos Software Patents could have on crystallography, and on those who perform it for the scientific and public good. It asserts the opinion that Software Patents (which in the technical literature can also take the title of "business method patents" or "*computer-implemented b*usiness method patents") are a "clear and present danger" to science and crystallography. "We cannot stand on the shoulders of giants if the giants wear spiked shoulder pads." At first glance, much of this topic may seem absurd, bizarre and a work of fiction. Thus it tries at every point to include multiple relevant web references so that interested readers can easily check and research it for themselves.

What are "Software Patents" and why are they different to normal Patents. The patenting of "Ideas" and "Scientific Knowledge" rather than "Inventions".

The main principle behind patents is *"not to protect inventors. The original aim is to ensure technological advances and their propagation by encouraging the publication of knowledge"* (http://objcryst.sourceforge.net/ and http://www.smets.com/it/policy/useright/). Patents are about "encouraging innovation" not "stifling it", and on the whole are meant to confer a social benefit by allowing a short term ownership of an invention, after which it becomes part of the public domain. Protecting the inventors via the use of patents is a means to that end. A good elaboration of this is described in Bruce Perens' "Software Patents vs. Free Software" at http://perens.com/Articles/Patents.html. An extract of which states:

"The U.S. is important to this discussion because it's the cradle of software and business-method patents. Most other nations don't allow such things to be patented, although certain companies are lobbying to change that. U.S. law, being a descendent of English Common Law, has preserved the patent, though in a more limited sense than its royal application. The Constitution of the United States states:

> *Congress Shall Have Power To [...] promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries*

This is the entire justification for the existence of the U.S. copyright and patent system. Both exist to promote the progress of science and the useful arts: technology, literature, and so on. If the patent and copyright system do not have the effect of promoting progress, they aren't constitutional."

The problem with the stereotypical Software Patent is that it tries to patent an "idea", which patent law is meant to forbid. Why should anyone care if "ideas", rather than "inventions", are being patented? Can't "ideas" and "inventions" be one in the same? While there are multiple lines of argument on this topic, the one considered most relevant for this text is that of the freedom to express ideas, scientific and algorithmic. John Goerzon gives an American perspective on why Software Patents have the great potential to be a serious problem in his text "Software Patents: Goliath Killing David?" (http://www.complete.org/news/952490640/, March 2002)

"What's so bad about that? Several things. First, we have a first-amendment right to be able to publish our own ideas as we see fit. Patents give the holder the right to restrict your ideas if your ideas too closely resemble theirs. This is different than copyright, which prohibits you from copying someone else's ideas, but lets you express your own.".

Summarising this in Patent Speak, Software Patents "stifle innovation". Summarising in relatively normal language, Software Patents stifle and restrict the free expression of ideas. They run contrary to democratic principles, and democratic principles of what constitutes a social benefit. In principle, Software Patents can apply to any piece of scientific knowledge that is algorithmic in nature (e.g., the ideas as implemented within crystallographic software). This would allow the owners of the ideas within Software Patent to restrict the spread of "similar" ideas via the legal tool of pursuing patent infringement actions against offenders.

Software Patent history. If they are just a US legal disease, why should Europeans Worry? And why Worry Now?

So the story goes, in the "good old days"™, the US Patent Office (PTO) did not recognise Software Patents. They were considered "ideas", not "inventions", and thus were not patentable. Unfortunately, the US Patent office became very under resourced and very badly managed. Between the mid 1980's to early 1990's, they started letting through Software Patents, mainly due to not having the expertise and resources to properly judge them. If in doubt, or no expertise was available, Software Patents were generally passed, no matter how trivial or how absurd. This also helped make the PTO management statistics on "numbers of patents approved" look "better". For more florid history and opinions of the US Patent office, including those of current and former USPTO employees, refer to Evan Ratliff's "Why many invalid patents are being issued by Patent Offices " (http://www.bustpatents.com/exam.htm); as well as "Patent examination system is intellectually corrupt" by Gregory Aharonian (http://www.bustpatents.com/corrupt.htm), which includes a statistical analysis on the US patent examination procedure.

The situation in Europe is very different. At present Software Patents are "officially" considered invalid. A thirty-three page document by Jean-Paul Smets "Software Useright: Solving Inconsistencies of Software Patents" (1999 - http://www.smets.com/it/policy/useright/useright.pdf) gives a useful overview on Software Patents, and their relevance to Europe. E.g., Section 2.1 on "New, inventive, industrial application", quotes:

"Article 52 of the Munich convention states that `European patents shall be granted for any inventions which are susceptible of industrial application, which are new and which involve an inventive step' and adds in paragraph 2 that the following in particular shall not be regarded as inventions:

1. discoveries, scientific theories and mathematical methods;
2. aesthetic creations;

3. schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;
4. presentations of information.”

To most crystallographers working on European soil, Software Patents were pretty much an irrelevancy; until recently that is. The problem now is there exists a powerful effort to bring American style Software Patents to not only Europe, but the rest of the world. Quoting from the EUROLINUX Alliance (http://swpat.ffii.org/);

“For the last few years the European Patent Office has, contrary to the letter and spirit of the existing law, granted about 30000 patents on computer-implementable rules of organisation and calculation (programs for computers). Now the European patent movement wants to change the law so as to legalise this practise and remove all barriers to patentability.”

Also refer to “CEC & BSA 2002-02-20: proposal to make all useful ideas patentable” - http://swpat.ffii.org/papers/eubsa-swpat0202/.

About the alarming claim that “about 30000” Software Patents have been passed by the European Patent Office. A French Language publication described on the Noepatents.org website titled “Droit de l'Informatique” is considered relevant (http://www.noepatents.org/reference/law.html):

“Recent editions of this textbook explain that, according to the Munich Convention, pure computer programme can not be patented but that the EPO has developed various tricks to circumvent the Law. The patents resulting from these tricks may not be valid in case of dispute.”

Quoting an extract from David Farber’s posting; “IP: Has the U.S. Patent Office really reformed?”(http://www.interesting-people.org/archives/interesting-people/200103/msg00075.html),

“Exporting American "Innovation" Abroad: While the U.S. PTO struggles to reform its examining procedures, and members of Congress propose potential solutions (such as the Business Method Patent Improvements Act of 2000), at least one other branch of the U.S. government -- the Office of the U.S. Trade Representative -- appears to have been laboring to ensure that other nations adopt the controversial American practice of granting patents to business methods. (USTR is the federal agency charged with negotiating and enforcing America's trade positions with other nations.)”

To put it mildly, anti-Software Patent activists in Europe are not enamoured by the idea of suddenly finding European software and “ideas” infringing against an avalanche of US Software Patents.

“Trivial” Software Patents: making the trivial sound complex and novel.

The main theme of this article is the problem associated with the patenting of ideas and scientific knowledge (via the use of Software Patents) versus the free expression of ideas and scientific knowledge. As a side issue, many Software Patents issued by the US Patent Office routinely involve “triviality” and “prior art”. Many Software Patents patent the trivial, but describe their claims in complicated “patent speak”. This can also help mask the patenting of an “idea”, as though it was really a novel, non-trivial “invention”. In another US based article, Mark Nelson states (“Patent Legislation Considered Helpful” - http://dogma.net/markn/articles/Patents/Patents.html):

“The past 10 years have seen a big change in the way programmers view the US Patent system. We've moved from our long-standing position of blissful ignorance into a keen awareness that each and every line of code we write just might step on the toes of an existing software patent. Times have certainly changed.”

A good resource on understanding triviality in Software Patents is Richard Stallman’s “The Anatomy of a Trivial Patent” at http://lpf.ai.mit.edu/Patents/anatomy-trivial-patent.txt. An extract from the introduction follows:

“Programmers are well aware that many of the software patents cover laughably obvious ideas. Yet the patent system's defenders often argue that these ideas are nontrivial, obvious only by hindsight. And it is surprisingly difficult to defeat them in debate. Why is that?

One reason is that any idea can be made look complex when analyzed to death. But another reason is that these trivial ideas often look quite complex as described in the patents themselves. The patent

system's defenders can point to the complex description and say, "How can anything this complex be obvious?"

I will use an example to show you how. Here's claim number one from US patent number <mark>5,963,916</mark>*, applied for in October 1996:"*

Included is a further extract of Richard's thoughts, converting the Software Patent's "complex" claim into "trivial" English ("Complex Claim" in italics, "Trivial English" in bold italics):

   * *c) choosing at least one pre-selected portion of the pre-recorded music products from the central host server;*

In other words, the user clicks to say which link to follow. That is typical for web servers; if they had found another way to do it, that might have been an invention.

   * *d) receiving the chosen pre-selected portion of the pre-recorded products; and*

When you follow a link, your browser reads the contents. This is typical behavior for a web browser.

   * *e) interactively previewing the received chosen pre-selected portion of the pre-recorded music product.*

This says that your browser plays the music for you. (That is what many browsers do, when you follow a link to an audio file.)

Richard continues in a separate section:

Now look at a subsequent claim:

3. The method of [149]claim 1 wherein the central memory device comprises a plurality of compact disc-read only memory (CD-ROMs).

What they are saying here is, "Even if you don't think that claim 1 is really an invention, using CD-ROMs to store the data makes it an invention for sure. An average system designer would never have thought of that."

The main practical rationale behind patents is to help justify and protect the often extremely high costs of research and development for new inventions (complex equipment, industrial processes, pharmaceuticals, etc). In this manner, patents exist to help encourage "innovation". The high cost of development is not a general attribute of a "Trivial" Software Patent. However, there are a large range of complicated software projects (Operating Systems, Office Suites, Graphics Suites, etc), involving many years and much expense to create. Yet in general these have been able to develop and thrive without the protection of patents (though they have been protected by Copyright and Trademarking). Restating this another way, patents have not been required to help encourage continued innovation in the field of software development. But as is being shown in the US, enforcing Software Patents has incredible potential as a weapon to destroy software development and innovation. A counter argument to the above is, that if there had have been a tradition of patent protection for software and algorithms, there would be even more stratospheric innovation in software. However, a paper by MIT and Princeton University researchers dispute this. They claim to answer the question "How could industries such as software, semiconductors, and computers have been so innovative despite historically weak patent protection?"(James Bessen and Eric Maskin, "Sequential Innovation, Patents and Imitation" – revised July 2002 - http://www.researchoninnovation.org/patent.pdf)

Some mischief, mayhem and chaos caused by Software Patents in the non-crystallographic world

Software Patents have been causing a goodly amount of mayhem in the non-crystallographic world, with effects that can go beyond their initial US jurisdiction. Web available references on this include: "Amazon sues Barnes & Noble over checkout system" (http://www.theregister.co.uk/content/archive/7437.html); "JPEGs are not free: Patent holder pursues IP grab" (http://www.theregister.co.uk/content/4/26272.html); "MP3 owners get stroppy with open source coders" (http://www.theregister.co.uk/content/4/19982.html); Software download patent refuses to die" (http://news.com.com/2100-1023-270021.html?legacy=cnet&tag=mn_hd) and "US patent mess will get worse before it gets better" (http://www.theregister.co.uk/content/archive/10119.html).

Be wary that many web based articles about Software Patents and their effects can suffer from being written mainly for programmers in "Geek Speek". Most web based authors consider Software Patents a threat to free speech and the free expression of ideas (and a threat to many livelihoods). Thus they do not make for "objective" reading and tend to be very negative and mocking of Software Patents. An American article by James Gleick ("Patently Absurd" March 2000 http://www.around.com/patent.html) gives a further indication of the chaos Software Patents are causing:

"Software and algorithms used to be unpatentable. Recent court decisions and patent-office rule-making has made software the fastest growing patent category, and companies are rushing to patent the most basic methods of doing business. "This is a disaster," says Lawrence Lessig, a Harvard law professor and cyberspace expert. "This is a major change that occurred without anybody thinking through the consequences. In my view, it is the single greatest threat to innovation in cyberspace, and I'm extremely skeptical that anybody's going to get it in time."

The litigation is spreading fast. Multi-Tech Systems has just sued the three leading PC makers, Compaq, Dell and Gateway, over patents on transmitting data over a communications line. A St. Louis patent broker is suing Yahoo over a "method of effecting commerce in a networked computer environment in a computerized system" — that is, shopping on line. Another Internet startup, Priceline.com, has patented its Internet version of an ancient auction technique, the name-your-price "reverse" auction, and is suing Microsoft's Expedia.com travel service. Microsoft, meanwhile, has infuriated much of the Internet community by patenting a well-known "style sheet" technology just as it was being adopted as standard by the World Wide Web consortium."

A hypothetical example of a crystallographic Software Patent and its possible effect on Crystallography

At this point we will create a hypothetical, "fantasy" example of Software Patents as could be applied to scientific crystallography. Let us look at the Crystals structure refinement suite as developed by the Chemical Crystallography Laboratory of Oxford University (http://www.xtl.ox.ac.uk/crystals.html). In reality, Crystals is copyrighted "2002, Chemical Crystallography Laboratory, University of Oxford, UK". Under this copyright, it is a set of ideas as implemented in software – a work of art. There is legally nothing to stop anyone else implementing similar ideas and similar works of art; such as Shelx, NRCVAX, GX, CAOS, Xtal, DS*System, etc. However, what would happen if the ideas based in the Crystals software were protected by Software Patents? Any program doing anything too similar (i.e., crystal structure refinement - which depending on how the Crystals Patent was written - may or may not be limited to the use of Least Squares, and possibly be inclusive of any diffraction phenomena) would now be potentially infringing the Crystals Patent. The owners of the Crystals Patent could then have the option of acting against this infringement of their intellectual properly; and thus let loose the dogs of law. A possible result would be that crystallographers with ideas similar to the Crystals Patent start seeking safer work in different careers, or become bankrupted by the legal proceedings against them. What if any of these software author ever admitted in their software manuals to using even just one of the ideas of Watkin, Prout, Carruthers, Betteridge or Cooper as implemented in Crystals? Under copyright laws, this is not a legal problem and normally consider laudable to acknowledge sources of ideas. But under patent laws, it would be a naïve, printed confession that will make the lawyers administering the Crystals Patent much easier.

Powder diffractionists would scoff at the above, and no doubt consider it a "single crystal problem". But on the creation and development of Rietveld structure refinement, they too find out in court that the Crystals Patent was indeed well written; its "patent speak" easily being interpretable as covering crystal structure refinement from powder diffraction data. Macromolecular crystallographers might be in the clear with their methods of structure refinement. However, if the owners of the Crystals Patent disagree, that would be for the courts to decide, not the scientists. As a real world example of Crystallographic "Patent Speak", US Patent 6,192,103 "Fitting of X-ray scattering data using evolutionary algorithms", issued on February 20, 2001, makes an adequate example:

"The present invention relates to X-ray metrology, and more particularly to the fitting of simulation models to X-ray scattering data for the purpose of determining parameters that characterize the structure of a material being tested." . . . "Examples of these methods include X-ray absorption, diffraction, fluorescence, reflectivity, scattering, imaging and fringe analysis. In the context of the present invention, the term "X-ray scattering" is employed as a generic term which collectively encompasses any known X-ray technique that is applied to materials characterization."

It is not unreasonable to expect the Crystals Patent would have been written to be as all encompassing as possible. Thus scientific progress is field of crystallography and related fields is now legally chilled by the cold fear of crystallographic programmers who dare not touch a computer keyboard or punch-tape device due to the risk of being sued. Oxford reigns supreme on monopolising these crystallographic ideas, and is unchallengeable for the time remaining on the Crystals Patent.

While this example is way over the top and potentially misrepresents many of the technicalities of patent law, it tries to display the potential for mayhem, mischief and chaos that Software Patents can achieve. It also tries to show some of the possible psychology of situations involving Software Patents - the patenting of "ideas". If such over generalising seems irrelevant to the rational, logical, real world system of the law; again let it not be forgotten that Software Patents are an American legal disease. This is the same legal system where someone can sue for such things as spilling a "scaldingly" hot cup of McDonald's coffee on themselves (winning in the process US$160,000 in compensatory damages and a "reduced" punitive award of US$480,000 (down from the initial US$2.9million award) – http://www.lectlaw.com/files/cur78.htm and http://www.atlanet.org/CJFacts/other/mcdonald.ht. The page "Patent Infringement Lawsuits: By the Numbers" at http://www.pearlltd.com/content/pat_inf_law.html is also worth browsing. *"Overall chances of success for the patent owner if the trial is held in Massachusetts and northern California, respectively: 30%, 68%"*. If the above is still considered too foolish to be taken even slightly seriously, some of the text by the "free software" author Bruce Perens ("Preparing for the intellectual-property offensive - Patents may be the ground on which the open source battle is won or lost at", http://www.linuxworld.com/linuxworld/lw-1998-11/lw-11-thesource.html) .

"One of the best defenses we have today is the fully disclosed nature of free software. If a patentable idea is used in free software first, and that software is distributed, we can use that software as evidence of "prior art" in overturning a patent. However, this is only a partial defense. The filer of a patent can claim that the invention was created up to a year previous to filing.

For example, AT&T informed me that it was filing a patent on the key principle in my "Electric Fence" malloc() debugging package almost a year after I first posted the source code to a Usenet newsgroup. The company claimed it had invented the principle before my posting. Although AT&T hasn't bothered me so far, there is a possibility that I will be hit with an expensive patent-infringement lawsuit sometime in the next 12 years or so. There's a good chance I'd win the suit, but I'd go broke in the process."

In general "Americans are not scared of the secret police, but they are of lawyers". Vincent Favre-Nicolin's GPL'd ObjCryst++ / FOX crystallographic software site at http://objcryst.sourceforge.net/ is relevant.

"What will you (V. Favre-Nicolin) do if software patents are made legal in Europe? As an individual I cannot take the risk of being prosecuted because some company patented a simple principle used in Fox or ObjCryst++ (use of integrated R-factors, the description of molecules using a Z-Matrix, genetic algorithms..). Therefore, I will still continue the Fox/ObjCryst++ development privately, but I will not be able to distribute it myself anymore."

To the uninformed reader, the above could appear very paranoid; who would bother trying to enforce such dodgy patented ideas against infringers? Though were you aware that the owner of US Patent 4,662,635, titled "Video game with playback of live events" (issued May 5, 1987), tried to sue the software maker of Duke Nukem 3D for his patented idea of incorporating animated video playback in video games (http://www.wired.com/news/print/0,1294,6252,00.html). And as shown above, were you familiar with US Patent 6,192,103 titled "Fitting of X-ray scattering data using evolutionary algorithms"? Or shown below, World Patent WO9906824 "Method and apparatus for determining molecular crystal structures"? These Patents were not known to Vincent when he added the above to his FOX/ObjCryst webpage on the 3rd of March 2002. They were only discovered during the writing of this document (Sep 2002). And there a many more Software Patents relevant to crystallography out there as will be elaborated below.

A Challenge to "Not Be Amazed"! : Some real Software Patents directly relating to Crystallography

In this and the following section are examples of software patents directly relating to crystallography. Of many potentially outstanding and amazing candidates, I have chosen the following to elaborate on.

US Patent 5,249,137

US Patent Title: Computer-aided chemical illustration system
Date of Patent: September 28, 1993
Abstract: A computer-aided chemical illustration system is disclosed. Techniques provided include: 1) efficient drawing of bonds; 2) drawing different bond types during a single mode; 3) determining bisect angles for bonds; 4) labeling atoms on the fly; 5) automatic alignment of atom labels; 6) custom alignment of atom labels; 7) changing the type, style, or orientation of an object while it is being drawn; 8) detection of ring structures; and 9) shifting bonds around on a ring.

Those in the know would conclude that the Assignee of this patent must be some optimistic, but highly ignorant and uneducated knuckle-dragging fringe dweller in the redneck backwoods of the American Deep South. Obviously, someone without the resources to pursue this patent, and only then against the extremely gullible. However, looking at US Patent Number 5,249,137 via the Internet (http://www.uspto.gov/patft/) will tell you that its "Assignee" is the Xerox Corporation of Stamford, CT, USA; with a reported yearly revenue in the tens of billions ($US). One might believe that it would be insane for a corporation to enforce such a Software Patent against the scientific and programming community. Surely, no corporation would ever be so stupid to do or try such a thing? However, this is not consistent with recent corporate history. For example, BT's (British Telecom) attempt to enforce its 1989 US Patent 4,873,662 titled "Information handling system and terminal apparatus therefor" (its ownership of the idea of Hyperlinks – the underlying basis of the World Wide Web) and obtain a "reasonable royalty" for those using the World Wide Web within the patent's jurisdiction. Surely it would be a waste of a patent, even something as dubious as a Software Patent, to not eventually try and enforce (when the time was ripe), or sell it to someone with a more involved interest. Otherwise, why would you bother having a Software Patent in the first place?

If you believe the above is outrageously misleading, you would be correct. The initial financial state of the holder of a Software Patent might not be that meaningful. The US oriented IPCreators provides much useful information to Patent Holders, including "Sources of Funding for Patent Litigation" (http://www.ipcreators.org/Enforcement/patfund.htm). Even for Patent Holders in the knuckle-dragging back-woods of the American Deep South, possible financial and legal assistance is well at hand to help enforce their Software Patents rights. The career option of being a Patent Troll has been very lucrative for some companies and some individuals ("Trolling for Dollars" - http://www.law.com/jsp/statearchive.jsp?type=Article&oldid=ZZZV4RVSSPC).

Companies can also stock-pile Software Patents for other "official" reasons, somewhat like nuclear weapons. They can be used as a defence against the different Software Patents of other companies; often involving the logic of MAD (Mutually Assured Destruction). From the European perspective, it might feel more like terrorist bioweaponry, of which the USA is its primary and unrestricted breeding ground. Mongol Hordes of US Software Patent holders, and not just in the USA, are itching to be able to press the button releasing this patent equivalent of a new wonder weapon on an unsuspecting, and potentially soon to be undefended world.

Some other Crystallographic Examples of Software Patents

There are many other examples of Software Patents and Software Patent applications directly applicable to Crystallography. Only "some" are listed below including an international Software Patent. A few are not strictly 100% Software Patents but are included out of special interest to the author, and might be better described as "business methods patents", which may still be considered as the patenting of an "idea". Unfortunately there is not enough space to list the patent "summaries", which can make equally remarkable reading. For instance US Patent 6,108,401 (Date of Patent: August 22, 2000), titled "Method of standard-less phase analysis by means of a diffractogram" which claims to be *"Using an estimate of the dispersive power of the individual atoms in the unity cells of the constituents"*. A casual glance of this patent could give the impression it tries to lay claim to many of the ideas behind conventional quantitative Rietveld analysis. These and many more Crystallographic Software Patents can be viewed fully on the web via a CCP14 based linking page at http://www.ccp14.ac.uk/maths/software-patents/.

US Patent Number: 5,200,910 (Date of Patent: April 6, 1993)
US Patent Title: Method for modelling the electron density of a crystal

US Patent Number: 4,811,217 (Date of Patent: March 7, 1989)
US Patent Title: Method of storing and searching chemical structure data

WO Patent Number: WO9906824 (Publication Date: 1999-02-11)
Title: Method and apparatus for determining molecular crystal structures

US Patent Number: 5,424,963 (Date of Patent: June 13, 1995)
US Patent Title: Molecular dynamics simulation method and apparatus

US Patent Number: 4,835,528 (Date of Patent: May 30, 1989)
US Patent Title: Cursor control system

US Patent Number: 6,438,205 (Date of Patent: August 20, 2002)
US Patent Title: System and method for reducing phase ambiguity of crystal structure factors

US Patent Number: 6,438,204 (Date of Patent: August 20, 2002)
US Patent Title: Linear prediction of structure factors in x-ray crystallography

US Patent Number: 6,326,619 (Date of Patent: December 4, 2001)
US Patent Title: Crystal phase identification

US Patent Number: 4,991,191 (Date of Patent: February 5, 1991)
US Patent Title: Quantitative analysis of the active table ingredient by powder x-ray diffractometry

US Patent Number: 4,592,082 (Date of Patent: May 27, 1986)
US Patent Title: Quantitative determination of mineral composition by powder X-ray diffraction

US Patent Number: 6,108,401 (Date of Patent: August 22, 2000)
US Patent Title: Method of standard-less phase analysis by means of a diffractogram

US Patent Number: 6,192,103 (Date of Patent: February 20, 2001)
US Patent Title: Fitting of X-ray scattering data using evolutionary algorithms

US Patent Number: 5,353,236 (Date of Patent: October 4, 1999)
US Patent Title: High-resolution crystallographic modelling of a macromolecule

US Patent Number: 6,411,676 (Date of Patent: June 25, 2002)
US Patent Title: Method for determining parameters of a unit cell of a crystal structure using diffraction

United States Patent Application: 20020107643 (Date: August 8, 2002)
US Patent Title: Process for pan-genomic determination of macromolecular atomic structures

US Patent Number: 5,448,498 (Date of Patent: September 5, 1995)
US Patent Title: Apparatus for generating a constraint condition in a molecular dynamics simulation

US Patent Number: 6,453,246 (Date of Patent: September 17, 2002)
US Patent Title: System, method, and computer program product for representing proximity data in a multi-dimensional space

United States Patent Application: 20020111761 (Date: August 15, 2002)
US Patent Title: Method for determining multi-dimensional topology

US Patent Number: 5,265,030 (Date of Patent: November 23, 1993)
US Patent Title: System and method for determining three-dimensional structures of proteins

As mentioned before, the language of patents can make it difficult for the non-expert to judge their scope; and thus who might be affected if rigorously enforced by legal action. It can be just as difficult to decipher whether they might be trivial; the degree of triviality; or whether they are describing ideas that were already obvious and/or in the public domain.

When Software Patent owners seek "relief"; and Patent Trolls Attack!

As the saying goes: *"The difference between 'involvement' and 'commitment' is like an eggs-and-ham breakfast: the chicken was 'involved' – the pig was 'committed'."* Imagine for the moment that you are "committed" to a European based macromolecular software suite. Commercial or freeware, it makes no real difference. And let us imagine that due to the pressure applied to rationalise patent laws globally, American Software Patents and US Patent rules become valid in Europe.

Might your macromolecular software be doing any protein modelling? Then it could be infringing US Patents 5,884,230 and/or 5,557,535, both titled "Method and system for protein modeling". Might it be generating electron density maps? If so, it might be infringing on US Patent 5,200,910, "Method for modelling the electron density of a crystal". Might it try analysing the rigidity of substructures within a macromolecule? Then it might be infringing US Patent 6,014,449 "Computer-implemented system for analyzing rigidity of substructures within a macromolecule ". Might it be trying to determine protein tertiary structure? If so, then it might be infringing US Patent 5,600,571, "Method for determining protein tertiary structure". Might it try and identify flexible parts of the macromolecule? Then it might be infringing US Patent 5,752,019, "System and method for confirmationally-flexible molecular identification". Might it try and minimise the bias in the electron density map? Then it might "soon" be infringing on the United States Patent Application 20020116133, "Method for removing atomic-model bias in macromolecular crystallography " Will your software be so bold as to display a protein structure? Then it might be "boldly" infringing US Patent 4,378,218 "Protein molecule model". Etc etc etc

Even if your crystallographic software is not affected by Crystallographic Software Patents, this still leaves non-crystallographic Software Patents that your program may be infringing. For example, most 3D displays of Electron Density Contour maps make use of a "marching cubes" algorithm; thus most likely infringing on US Patent 4,885,688, "Minimization of directed points generated in three-dimensional dividing cubes method". Many crystallographic programs use Fast Fourier Transforms, which might be infringing on US Patents like: US Patent 6,434,583; "Fast fourier transform apparatus and method" (issued August 13, 2002); and/or US Patent 5,371,696, "Computational structures for the fast Fourier transform analyzers "(issed December 6, 1994); and/or US Patent 6,430,587, "Method and apparatus for FFT computation" (issued August 6, 2002), and/or US Patent 6,058,409, "Computation apparatus and method" (issued May 2, 2000), etc. Just searching from 1996 to present (Sep 2002), there are 209 US Patents that mention "FFT" and 436 US Patents that mention "Fast and Fourier and Transform" Other similar examples could almost go on ad-infinitum.

As an extra observation, the above FFT patents makes the following article seem quite confusing: "The FFT: Making Technology Fly" by Barry Cipra (http://www.siam.org/siamnews/mtc/mtc593.htm), of which an extract follows:

"Cooley takes pains to praise the Gentleman-Sande paper, as well as an earlier paper by Sande (who was a student of Tukey's) that was never published. In fact, Cooley says, the Cooley-Tukey algorithm could well have been known as the Sande-Tukey algorithm were it not for the "accident" that led to the publication of the now-famous 1965 paper. As he recounts it, the paper he co-authored with Tukey came to be written mainly because a mathematically inclined patent attorney happened to attend the seminar in which Cooley described the algorithm. The lawyer, Frank Thomas, "saw that it was a patentable idea," Cooley explains, "and the policy of IBM and their lawyers was to be sure that nobody bottled up software and algorithms by getting patents on them." A decision was quickly reached to put the fast Fourier transform in the public domain, and that meant, in part, publishing a paper. "

However, if challenged by US Software Patent infringements that looked "trivial" or involved "prior art", you would still require the legal and financial resources to fight the Patent Trolls. "Software patents: an industry at risk" by Gordon Irlam and Ross Williams (http://lpf.ai.mit.edu/Patents/industry-at-risk.html) at this point makes extra and relevant reading.

A question to be answered is "what benefits accrue from Software Patents?" Well, assuming the ideas in your software infringe on at least one Software Patent (it seems pretty much impossible not to), then Software Patent owners can try and seek "relief" against companies, individuals and academics "infringing" on their ideas. The following quotes refer to "Patent Infringement and Relief for the Patent Owner "by Richard V. Westerhoff (http://www.tms.org/pubs/journals/JOM/matters/matters-9201.html). The ins and outs look rather convoluted, so again the following extracts try to get a feel for the generalities, not the labyrinth of possible caveats.

"In the United States, there are several forms of relief available to the patent owner who has successfully proven patent infringement. Such relief can include protection from future infringement; compensation for past infringement; certain of the costs of litigation; and, in exceptional cases, attorney's fees and increased money damages. As each of these forms of relief, or remedies, addresses a different injury, several or all may be applied in a single case" . . . "The remedy against continued infringement of a patent is an injunction. In a patent infringement suit, an injunction is a court order prohibiting the manufacture, use, or sale of the patented invention. This can include prohibition of the continued use of articles made prior to the issuance of the patent." . . . "Only one valid claim of a patent need be infringed to entitle the patent owner to relief. This is true even if other claims in the

patent are found to be invalid, as long as there was no deceptive intent on the part of the patent owner which led to invalidity."

Also, as pointed out in "Patent Infringement Lawsuits: A Brief Overview" at http://www.lawnotes.com/patent/lawsuit.html :

"A patent owner that proves infringement is entitled to the following (assuming that the accused infringer does not prove that the claim is invalid or otherwise unenforceable): i) damages not less than a reasonable royalty; ii) any profits proved to have been lost by the patent owner as a result of the infringement."

Some academics may be of the opinion that as they are not making any money from their software, they will be free to infringe these Software Patent; continuing their software development under the ideals of "socialist competition". A non-expert interpretation of the above is that adequate scope exists for Software Patent holders to prosecute free-for-all academics writing scientific and crystallographic software.

However as claimed in Don Lancaster's "The Case Against Patents" (http://www.tinaja.com/glib/casagpat.pdf); under the topic of **"Fact: Prior art is not needed to bust any patent"**, it may not be all the individual Patent Owner's way. While this article is not focussed on Software Patents, but individuals who might be trying to patent their inventions to make money; Software Patents holders who might be tempted to enforce their claims in court may want to take special note of risks described by Don of a "frivolous litigation countersuit".

"All you really have to do is show that the claims would have been reasonably obvious to any "practitioner in the field." That's all it takes.

As is often the case, a patent search gets made without actually looking at any of the non-patent history of the field in the way of key papers, seminars and trade journals. All you have to do is find someone somewhere that says it sure would have been obvious to them.

When (not if) your patent ends up busted, you will also run the risk of a frivolous litigation countersuit. Thus, if you have the temerity to try and defend your patent, you could end up being fined thousands of dollars."

Patent horror stories of similar ilk are given in Don's article, Patently Horrible at http://www.tinaja.com/glib/patnthor.pdf. This might give the impression that patents mainly benefit organisations with money and time to burn on very good lawyers - in defending patents and/or busting patents.

What can you do about the threat of Software Patents? Forewarned is Forearmed! (to some extent)

Because the traditions, logic and the social desirability of patents in a democratic society go against the patenting of "ideas", reading up on the history and potential mayhem Software Patents can be valuable. Forewarned can be Forearmed! On hearing that an organisation has applied for, or received Software Patents, this is a good reason to stay very alert, and not fall asleep in the back row of a conference theatre. One suggestion would be to treat organisations that enforce Software Patents as if they were Plague Carriers of Medieval times (in this case carrying an American legal form of it), and avoid having any business to do with them. While some potentially relevant Patent Numbers and Titles are provided with this text, their Assignees can be found on the US Patent Office and European Patent Office websites (http://www.uspto.gov/patft/ and http://www.european-patent-office.org/espacenet/info/access.htm). Companies and Institutes may try and point out the large cost of software development requires protection by the use of Software Patents. But Copyright and Trademarking can, and do, provide adequate protection for software. The patenting of ideas, such as found in Software Patents, if successful, mainly allow the threatening, destruction and suppression of others in implementing similar scientific ideas.

As Software Patents have the potential to cause considerable problems with UK and European businesses, your local MP may not be disinterested on this issue. Lists of UK Parliamentary Members are viewable at http://www.parliament.uk/directories/hcio/lists.cfm. There is also a "Fax a UK MP" website at http://www.faxyourmp.com/. Contact details for UK members elected to the European Parliament is provided at http://www.europarl.org.uk/uk_meps/MembersMain.htm.

As is also stated below, good set of very blunt resources for patents in general are visible via Don Lancaster's Patent Avoidance Library Tutorials at http://www.tinaja.com/patnt01.html#tutorials. Don is a "Midnight Engineer" (an inventor). His conclusion of trying to use the patent system for protection of conventional inventions is "There is not one patent in a thousand that cannot be busted by a diligent enough search for prior art done in obscure enough places." ("Busting a $650 Patent" at http://www.tinaja.com/glib/bustpat.pdf). Again, this might give the impression that patents mainly benefit organisations with money and time to burn on very good lawyers - in defending patents and/or busting patents.

European Groups Advocating Against Software Patents

Active politicking and activity against software patents of this type has claimed some effect on governmental decision makers. However, this should not be equated with the defeat of those trying to implement Software Patents in Europe. The "Petition for a Software Patent Free Europe" is available via the web (http://www.noepatents.org/) of which the present number of signatures (Sep 2002) are 125,390. People can still sign via the web as well. The other main anti Software Patents site in Europe is the EuroLinux Alliance (http://eurolinux.ffii.org/). Though based on a EuroLinux Alliance press release, this Petition may be having little effect on the European Commission (http://petition.eurolinux.org/pr/pr17.html). However, quoting from a UK Patent Office study document of March 2001 at http://www.patent.gov.uk/about/consultations/conclusions.htm, it initially notes under the title of "The Role of Patents" that "*Patents are a long-established means of encouraging innovation. A patent confers on its holder a legal monopoly, for a limited period, in commercial exploitation of the patented invention.*" On the issue of patenting "*computer-implemented* business methods" (Software Patents) it continues:

"There is no sign, at least to date, of a want of innovation in computer-implemented business methods, and nor was there in the US before business methods became patentable in 1998. Intense innovation has characterised this field. The Government's conclusion is that those who favour some form of patentability for business methods have not provided the necessary evidence that it would be likely to increase innovation. Unless and until that evidence is available, ways of doing business should remain unpatentable."

"Protesting" a Published US Patent Application

The option of challenging US Patents or Patent Applications is rarely highlighted on examined anti-Software Patent web resources. It might be concluded this is not considered effective against software patents (possibly due to the number of Software Patents being a large flood amongst an indifferent US Patent Office?). The following information is never-the-less included as an informational starting point. However, if similar to the European Patent System, challenging even the most ludicrous US Software Patent may be well beyond the means and resources of individuals and academics.

The US Patent Office webpage on "General Information Concerning Patents" (http://www.uspto.gov/web/offices/pac/doc/general/) states :

"Most patent applications filed on or after November 29, 2000, will be published 18 months after the filing date of the application, or any earlier filing date relied upon under Title 35, United States Code. Otherwise, all patent applications are maintained in the strictest confidence until the patent is issued or the application is published. After the application has been published, however, a member of the public may request a copy of the application file. After the patent is issued, the Office file containing the application and all correspondence leading up to issuance of the patent is made available in the Files Information Unit for inspection by anyone, and copies of these files may be purchased from the Office."

Under US Patent Law, there are options for the public to challenge a patent application under *37 CFR 1.291 Protests by the public against pending applications* (http://www.uspto.gov/web/offices/pac/mpep/documents/1900.htm). Quoting from this:

"1901.01 Who Can Protest
Any member of the public, including private persons, corporate entities, and government agencies, may file a protest under 37 CFR 1.291. A protest may be filed by an attorney or other representative on behalf of an unnamed principal since 37 CFR 1.291 does not require that the principal be identified."

Other content includes in section 1900 includes: "1901.02 Information Which Can Be Relied on in Protest"; "1901.03 How Protest Is Submitted"; "1901.04 When Should the Protest Be Submitted". As stated in 1901.02, "prior art" is not the only issue that a patent application can be challenged under.

Nicholas Groombridge and Christopher Loh note in "Congress Takes Aim at Business Method Patents" (March 6, 2001 – within http://www.weil.com), by tradition, challenging a patent application has not been common.

"Although current law permits the filing of such protests and petitions, see 37 C.F.R. 1.291 and 1.292, these provisions were almost never used, in part because U.S. patent applications traditionally were not published and therefore it was difficult or impossible for a third party to know what was pending before the PTO."

It is now possible to search for Published US Patent Applications published since 15 March 2001 via the web at http://www.uspto.gov/patft/. In theory, the Internet now makes it easy to be informed of Patent Applications, and evaluate what might be worth protesting.

Under, "1901.03 How Protest Is Submitted" (http://www.uspto.gov/web/offices/pac/mpep/documents/1900.htm#sect1901.03), it states

"Protestors are encouraged to use form PTO-1449 "Information Disclosure Statement" (or an equivalent form) when preparing a protest under 37 CFR 1.291, especially the listing enumerated under 37 CFR 1.291(b)(1). See MPEP § 609. In addition, the protest and any accompanying papers must either (1) reflect that a copy of the same has been served upon the applicant or upon the applicant's attorney or agent of record; or (2) be filed with the Office in duplicate in the event service is not possible."

The US Patent Office forms page is at http://www.uspto.gov/web/forms/

Obtaining a "Reexamination" of a US Patent

According to the US Patent Office website, there are also options "reexamination" of a patent in terms of challenging its validity. Be wary that in searching for US Patents, the present default US Patent Office web interface searches the years of 1996-2002. To search all years, you have to click on the list box "Select Years", and select "All years". If you don't do this, patents prior to 1996 will not be found and listed. (refer Figure 1) The page on "General Information Concerning Patents" at http://www.uspto.gov/web/offices/pac/doc/general/, while pointing out that *"Once the patent is granted, it is outside the jurisdiction of the USPTO except in a few respects"*; states, under "Correction of Patents":

"In a different type of proceeding, any person may file a request for reexamination of a patent, along with the required fee, on the basis of prior art consisting of patents or printed publications. At the conclusion of the reexamination proceedings, a certificate setting forth the results of the reexamination proceeding is issued."

Again, US Patent Office forms are available on the web at http://www.uspto.gov/web/forms/, which include: PTO/SB/57 Request for Ex Parte Reexamination Transmittal Form and PTO/SB/58 Request for Inter Partes Reexamination Transmittal Form.

There is also a Software Patents oriented webpage titled "Patents declared invalid by the PTO or the courts", with the reasoning behind the decisions at http://www.bustpatents.com/invalid.htm. Some patents that "should" be invalidated are also listed. This includes "US Patent 5,533,051: Method for data compression". This "should be invalidated" by the PTO (or the courts) as it would seem that "*patent 5533051 claims something mathematically impossible, and therefore should not have been issued.*" In the original posting, Jean-loup Gailly points out *"It took three years to the patent office to ascertain the validity of such a patent. A person with basic knowledge in mathematics and data compression can find the flaws immediately upon first reading."*
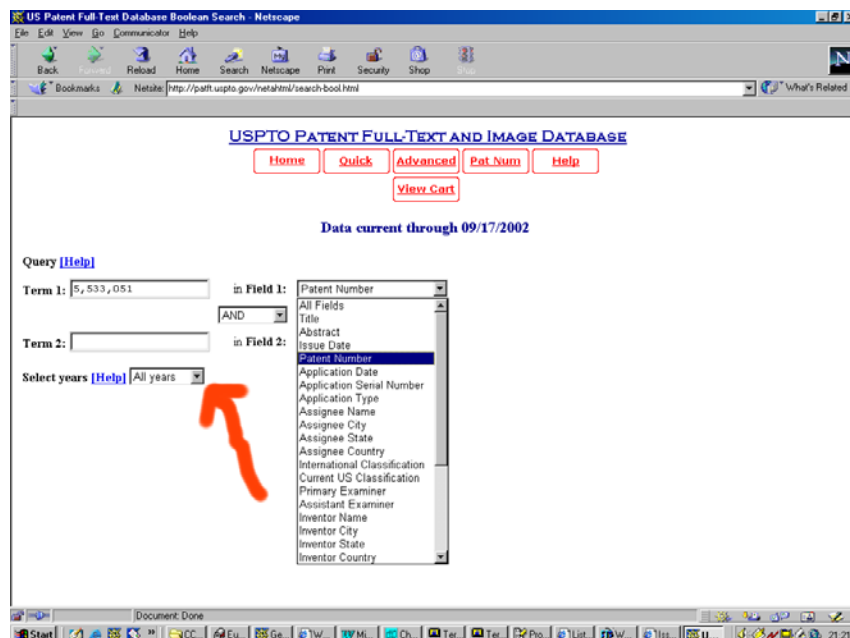
Figure 1: Example of searching for patents using the US Patent Office web interface via http://www.uspto.gov/patft/. The arrow points to the "Select years" option. For finding patents beyond the default years of 1996 to the present (Sep 2002), you should select the "All years" option. In this example, we are searching for patent 5,533,051. If you wished to find all the patents that mentioned the word crystallography, you would type "crystallography" into the "Term 1" field; select "All Fields" in Field 1"; then click the "Search" Icon (obscured in this image by the "Field 1" options. At time of writing (Sep 2002), this resulted in 3,737 hits. The results webpage allows the option of "refining" the search. Thus retrying by using "crystallography and software" in "All fields" then shows 982 hits. Using "crystallography and powder and diffraction and software" shows 115 hits. Please bear in mind that patents of possible interest might only be found using unexpected keywords. However there is useful extra help in the patent output page in the form of hyperlinks to cited patents, as well as patents that reference the displayed patent. Thus finding one relevant patent can lead you to a useful forest of related patents via the USPTO web interface. For instance, US Patent 6,192,103 "Fitting of X-ray scattering data using evolutionary algorithms" is referenced by hyperlink to another US Patent; 6,453,002 "Differential measurement of X-ray microfluorescence", which then quickly leads into a plethora of other patent citations, some relevant as crystallographic software patents, some not. While the competence of the US Patent office may be called into question over their issuing of Software Patents, their web based patent searching facility is brilliantly implemented.

Sneaking Software Patents in via the European Patent Office or the WIPO Patent Cooperation Treaty (PCT)

While nearly all the above information concentrates on using the US Patent system for patent searching, this would miss other Crystallographic Software Patents issued via the European Patent Office or by WIPO (though as previously cited, EPO Software Patents would most likely be found to be invalid if challenged). One such Software Patent outside the US Patent database system is the previous mentioned WO9906824; "Method and apparatus for determining molecular crystal structures" (Publication Date: 1999-02-11). This makes claims on ideas for solving molecular structures from powder diffraction data. At present, the European Patent Office website seems to offer the best Internet method for discovering PCT (and European) Patents (http://ep.espacenet.com/). If you have not heard about international patents before, refer to the World Intellectual Property Organistion (WIPO) – http://www.wipo.org/; *"The WIPO-administered Patent Cooperation Treaty (PCT) provides for the filing of a single international patent application which has the same effect as national applications filed in the designated countries.".* Personal experience of using the EPO website is that it is very inferior to the US Patent Office web interface in terms of being able to search and find relevant patents. It can often be quite neurotic and return zero matches even if you know the Patent Number, title or authors.

At time of writing, there seems to be no method of finding European of PCT "published patent applications" via the web. Poor web implementation compare to the US Patent Office system could mean using the WIPO or European Patent System may be considered a stealthier way of minimising

the chances of Software Patent applications gaining the public eye. Thus it minimises the risk of Software Patents being found by people who may have grounds for protesting or challenging them.

The impracticality of academics and individuals challenging ludicrous European Patent Applications

In theory, it looks like concerned individuals could easily challenge these ludicrous patents on issues such as "prior art", "obviousness to a practitioner", "inventive step", etc. However, in practise, it does not look practical or possible for a concerned individual or academic to challenge a European Software Patent application. Professional comment received on this topic can be summarised as:

- *The challenge must be done no more than 9 months after the patent has been granted*
- *The time required for the opposition process goes from 1 to more than 2 years after filing the opposition. The last step is a hearing in Munich.*
- *You need a specialist recognised by the European patent office (which requires a special qualification) to represent you in front of the EPO. (The organisation passing on these comments mentioned they send 2 people to Munich; one knowing all technical details, the other being the legal representative).*
- *The total cost is in the order of 10,000 to 15,000 Euros*

Finally it was also stated that *"opposing a European patent was a two-edged sword, because if the patent is not "entirely" invalidated, it strengthens the remaining part of the patent".* Thus it would seem that people applying for ludicrous European Software Patents need not fear opposition from knowledgeable and motivated individuals.

Advice for "Inventors" who might consider applying for Patents - from one who claims to know

For individual "inventors" who believe Patents are the thing to own and enforce, an article worth reading is within Don Lancaster's "Patent Avoidance Library" (http://www.tinaja.com/patnt01.html) titled "The Case Against Patents" (http://www.tinaja.com/glib/casagpat.pdf). This may be especially valid for individuals considering whether to apply for Software Patents.

On the topic of patents, Don gives a number of very blunt comments, including:

"All a patent does is give you the right to sue someone in a civil action. At some future date in a ridiculously costly, extremely drawn out and easily circumvented legal process. Nobody has ever "won" any patent litigation. The main purpose of patent fights are to cause more grief and harm to the opposition than you are causing yourself. Almost always, this purpose fails miserably."

On the topic of "Some Better Alternatives" (to Patents), Don offers some advice to "individuals":

"So, if you are a Midnight Engineer, just what are the alternative methods for successfully marketing your ideas and concepts? Based on many years of personal experience and several cubic yards of overflowing third-party patent victim files, here's what I'd suggest . .

First, totally avoid any and all contact with anything even remotely patent related. In any way, shape or form.

Do so religiously.

Second, don't even bother creating anything in any field in which you are not eventually certain to become an expert. An expert that is thoroughly familiar with the technical literature, the history of the field, the marketing realities, the insider trade journals, and the mainstream tools and techniques in use. There is no point whatsoever in writing forest fire simulation software if you have never sharpened a Pulaski. Nor (as sadly happened to yet another victim just this morning) in patenting a "new scheme to replace inductors" without having read and understood Sallen and Key in their 1955 classic paper.

Third, publish all your key secrets and ideas in a major magazine, leaving out no detail, and omitting no insider secrets. This immediately can generate positive cash flow for you and safely tucks all your ideas away in the public domain, preventing most others from attempting to patent them. This also will expose your new ideas to the widest possible audience.

Fourth, try to set up some royalty arrangement with a smaller to medium firm in some position to market and distribute your invention. The normal royalty payment is typically in the five percent range. Now for the tricky part: They must come to you, and never vice versa. That is why it is super important to publish your ideas and creations and expose them as widely as possible.

You normally have one and only one defense against getting ripped off in any royalty setup: The expectation that you will be delivering newer and better stuff in the future. And that is all.

Fifth, employ the shotgun technique. There is no way that one single idea or product will hack it. To survive in this game, you'll need hundreds or even thousands of new ideas and concepts working for you on a total lifetime and total lifestyle basis. Chances are that one or two genuine winners will pay for all the others lost or stolen.

Finally, be realistic. You don't create things to get filthy rich. You create things because you like to create things and have some compelling desire or need to do so. As long as there are enough nickels to keep going, that is all that should really matter."

A good selection of articles by Don are viewable via Patent Avoidance Library Tutorials at http://www.tinaja.com/patnt01.html#tutorials. These include "Busting a $650 Patent" (http://www.tinaja.com/glib/bustpat.pdf). As has been stated twice before, the above might give the impression that patents mainly benefit organisations with money and time to burn on very good lawyers - in defending patents and/or busting patents.

Summary

This article asserts the opinion that Software Patents (i.e., the patenting of "ideas") are a "Clear and Present Danger" to Crystallography, and the Crystallographic software that makes it function and progress. The author is not a legal expert, so it would not hurt to validate the above via other means, which is why web links and key words are given at every opportunity. Given that "forewarned is forearmed", it is the author's option that Software Patents are, at minimum, an issue that crystallographers and anyone using the fruits of the ideas embedded in software need to be aware of. Crystallographers need to be aware that Software Patents abuse the patenting system by allowing the owning of "ideas" (as distinct from "inventions"). In principle, Software Patents can be applied to any piece of scientific knowledge that is algorithmic in nature. Thus Software Patents allow their owners to try and legally restrict and suppress the expression of similar scientific and programming ideas by others. The above set of references, including a CCP14 page at http://www.ccp14.ac.uk/maths/software-patents/ should give the reader a set of reading resources that can allow exploration of this issue. This includes pointers to pro Software Patents websites.

Crystallographers might seek to reassure themselves that they are too far removed from the real world to be at risk of being seriously touched by Software Patents; perhaps mirroring analogous thoughts of the civilian and refugee populations of Dresden prior to February 13th 1945? Surely crystallography is not wealthy enough, or high profile enough to attract the attention of the "Patent Trolls"? Would a company's lawyer ever debase him or herself to take interest in a lowly Crystallographer doing lowly Crystallography? Given the present existence of software patents directly relevant to crystallography, and verbal folklore about what some company lawyers have done in the past to some crystallographers, this looks very wishful thinking. To restate this more forcefully: as shown above, multiple Crystallographic Software Patents exist. By definition, this means the Crystallographic Software Patent Trolls are already "out there".

It seems appropriate to end with a quote attributed to Martin Niemöller (1892-1984), a German Lutheran theologian and pastor, who from 1937 to 1945 was imprisoned in various concentration camps such as Sachsenhausen and Dachau; within a legal system that very much restricted the free distribution of ideas.

"In Germany they came first for the Communists and I didn't speak up because I wasn't a Communist. Then they came for the Jews and I didn't speak up because I wasn't a Jew. Then they came for the trade unionists and I didn't speak up because I wasn't a trade unionist. Then they came for the Catholics and I didn't speak up because I was a Protestant. Then they came for me--and by that time no one was left to speak up."