

Patent search

Alexis Grandemange (alexis.grandemange@pagebox.net)

Motivation

In the past you could start a software business without knowing anything about patents. This is no longer the case. At the end on the 90 s it seemed possible to patent almost any kind of process or method. This is no longer the case. The current software patenting system favors corporations, which can support the cost of building patent portfolios and dedicate more resources to infringement disputes: legal departments of companies like IBM or Microsoft are bigger than most software companies. Furthermore while patents exist to reward innovation, the software industry provides numerous examples of blatant copy, which seems to show that the patent system fails to achieve its purpose. One reason is that a credible software patent deters other companies of using its method or process. If the patentee has no follower, if the patented function is not trendy customers do not buy the product and the patented function has no commercial value. Therefore quite often a company tolerates first what it complains about later on. To learn more about this aspect you can read [Society and Computing](#).

However patenting is not the only way to protect your intellectual property. The patent system makes increasingly difficult to keep secret your intellectual property but at the same time the technological evolution makes easier and increasingly effective to publish or donate this intellectual property.

In this document we explain what a patent is and we present the patent system with its national patent offices and its treaties. We discuss software patents pending issues and we present the different Intellectual Property strategies. This document can also help for market intelligence. Patents are usually a reliable source of information. The document alternates the explanations, examples and links (including all pages from where we borrowed ideas and sentences) needed to understand the matter. We also introduce a method and a set of Open Source tools to help finding patents and prior art.

The original version of this document was published in December 2002. The November 2003 version and the current October 2004 were major updates.

We wrote another [document](#), called Business Method and Software patents, in which

1. we consider in more detail technical aspects,
2. we try to define what an invention is,
3. we present an innovation assessment method, and
4. analyze the MercExchange v. eBay case.

Jargon

We avoid as much as possible to use legal language. However we quote and link to legal documents. You also need to understand this language to read USPTO manuals. Some words are old English, some are French but

most of them come from the Latin. <http://www.users.globalnet.co.uk/~loxias/legal.htm> is entertaining. You can also look at <http://www.mylawyer.com/glossary.htm> for a glossary.

Patents

In this section we briefly explain:

1. Why patents exist
2. Why software patents exist
3. What is patentable

Why patents exist?

Patents exist to grant inventors "the right to exclude others from making, using, offering for sale, or selling" inventions. The idea is to provide an incentive for inventing, the inventing process being perceived as an expensive and risky business. The American constitution says: "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."

That is arguably true for some inventions:

In 1902, a patent was granted in the United States to William Heal on the float glass process. Heal described the feasibility of continuously melting and delivering molten glass onto molten tin in a chamber, and then drawing the sheet of glass along the surface of tin into an annealing lehr. This concept was not actively pursued until the 1950 s when Pilkington Brothers in Great Britain initiated an active program to develop a float process for commercial use. The process turned to be so successful that Pilkington competitors had no choice but to use the process and during a while most of the Pilkington revenue came from license fees.

The idea was known of everybody. Pilkington was the only company to take the risk to invest on the concept up to the point it was effective. The very reason why patents exist is to reward and encourage that sort of attitude.

If you want more information you can read the [FAQ](#) of the Office of Intellectual Property Management of the University of New Hampshire.

Why software patents exist?

Software patents exist because software programs implement processes and methods and because these processes and methods can be patented. The things are however subtler: According to the American law it should be possible to patent "anything under the sun that is made by man". However it is not possible to patent mathematical algorithms because they are the expressions of natural laws that are merely abstract ideas constituting disembodied concepts or truths that are not useful. See http://cyber.law.harvard.edu/is98/final_papers/Lee.html for more information: algorithms that do not

manipulate abstract ideas can be patented.

The idea is that we can patent human inventions but not natural laws. Einstein could not patent the relativity whereas Bell could patent the telephone. In the past it was easy to distinguish between natural laws and inventions though a natural law is indeed a human invention that turned to be the simplest explanation for a set of observations. In the future the simplicity criteria could be replaced by usefulness for a given set of applications and different explanations to observations could co-exist. Therefore the patentable domain may broaden again.

Patentability conditions

- **New or Novel:** The invention must be demonstrably different from publicly available ideas, inventions, or products (so-called "prior art"). This does not mean that every aspect of an invention must be novel. For example, new uses of known processes or machines are patentable. Incremental improvements on known processes may also be patentable.
- **Useful:** The invention must have some application or utility or be an improvement over existing products and/or techniques.
- **Non-obvious:** The invention cannot be obvious to a person of "ordinary skill" in the field; non-obviousness usually is demonstrated by showing that practicing the invention yields surprising, unexpected results.

Most software patents are filed by firms. In an Empirical Look at Software Patents, James Bessen and Robert Hunt conclude; "U.S. legal changes have made it easier to obtain patents on inventions that use software. Software patents now comprise 15% of all patents, mostly acquired by manufacturing firms and large firms. Only 6% belong to software publishers. Our regression analysis finds that these patents differ from others: they are cheap patents." They also explain: "Firms that acquire the largest share of software patents appear to be just those firms that acquire large portfolios of patents for strategic purposes."

Most software patents describe a process implemented in a project or in a prototype. Because they did not require a specific R&D effort such software patents are cheap. The project was funded only to fulfill a marketing request and designers of "ordinary skills" made reasonable design choices, which allowed a successful implementation. As a result:

1. A company does not pay its designers and programmers to identify surprising and unexpected results. Therefore the invention is usually obvious for a person of the art.
2. The invention is new for the first company that chooses to develop a product with some new functions. The patented process is the most obvious means to offer the new functions.
3. The inventors do not aim to be granted the right to exclude others from using the same process or method. The inventors aim to be granted the right to exclude others from releasing products with the same functions.

We have shown in this section that Intellectual Property is a rapidly evolving area. Because it must remain consistent for all invention domains and in time, Intellectual Property becomes increasingly complex.

Practices

A software patent has to be written in terms that someone that never heard about computers can understand. To the opposite of biotech patents there are no formal coding rules.

An interesting aspect of software development is that this is a cross-domain activity: an invention can serve many needs almost as many needs as the hardware that run the programs. However patent offices usually follow another way. They classify inventions according to their claims AND to the business activity where the invention was first applied. This approach reduces the scope of the protection granted by the patent regardless of the novelty and usefulness of the patent. It makes patent scattting possible: someone who knows two business domains can file a patent in domain B with a process of domain A: the patent is a new use of a known process.

In this section we present the existing classifications and we comment an example.

Classification

The two most common classifications are the United States Patent and Trademark office (USPTO) and the World Intellectual Property Organization (WIPO) classifications.

You can find the USPTO classification at <http://www.uspto.gov/go/classification/> and the international classifications at <http://www.wipo.int/classifications/en/index.html>.

The most useful parts for the beginner in the USPTO page are:

- A search US Manual of Classification tool
- The link to the Class numbers and titles

On the WIPO page select the International Patent Classification. You will find eight sections:

- SECTION A HUMAN NECESSITIES
- SECTION B PERFORMING OPERATIONS; TRANSPORTING
- SECTION C CHEMISTRY; METALLURGY
- SECTION D TEXTILES; PAPER
- SECTION E FIXED CONSTRUCTIONS
- SECTION F MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING
- SECTION G PHYSICS
- SECTION H ELECTRICITY

Computers are classified in Physics like tobacco is classified in human necessities.

Software is classified in Electric Digital Data Processing (G06F) though software processes are usually not linked to a particular hardware model: an optical computer implementing a Von Neuman machine could run them.

We describe later in this document how to find the classes of interest for your business domain.

Example

Many articles have been written on the case described in this section.

Our intent here is to give information and not to comment decisions or acts.

Background

CRSs, also known as Global Distribution Systems (GDSs) were invented at the end of the 60 s mainly to serve air transportation needs. The air transportation industry comprised airlines and travel agencies. Travel agents sold the tickets whereas the airlines provided the transportation means. The CRS idea was to connect the travel agents to an intermediate system connected to the airline systems. The concept was similar to Internet online sell today. When we exclude the technology available at design time from the scope we find that CRSs and Internet business use the same processes:

1. In both cases a large number of devices are connected to the same machine cluster. Only protocol and network topology differ.
2. In both cases a software program handles user requests, a transaction monitor in case of CRSs, a HTTP server in case of Internet business.
3. CRSs activity implies the exchange of messages with airlines like Internet business implies exchanging messages with partners for Credit card processing and delivery. CRSs use Host-to-Host communication whereas Internet Web service uses TCP. Quite interestingly CRSs and airlines were early adopters of the EDIFACT standard when XML was designed to facilitate online business and B2B.

The travel agent equipment is different in two ways from end-user equipment:

- The travel agent has a special printing facility. Because air tickets have a value, they must be issued only once. This requirement has an impact on the printer hardware that must be more reliable and report incidents and on the CRS system that must handle these incidents.
- The travel agent has a card reader to accept payment

An inventor called Lawrence B. Lockwood filed a set of patents:

1. 4,359,631 filed in 1980 and reissued in 1986 as RE32,115 that describes a self-service terminal
2. 4,567,359 filed in 1984 that describes an automatic information, goods and services dispensing system
3. 5,309,355 filed in 1993 that describes an automated sales system

4,359,631/RE32,115 "relates first to automatic reservation and ticketing terminals; second, to programmed audio-visual displays used in connection with the sale of goods and services, and to automatic, around-the-clock service tellers" and describes something similar to the self service terminals we find now in

Patent search

airports and railway stations, to the ATMs and even to the good dispensers.

4,567,359 "relates to a system for automatically dispensing information, services and products to customers in a self-service fashion. Such a system may be used, for example, for automatically dispensing insurance quotations and policies". 4,567,359 extends the CRS concept to other industries.

5,309,355 "is directed to an automated sales system. Specifically this invention is directed to a tool for augmentation of sales and marketing capabilities of travel agency personnel in conjunction with computerized airline reservation systems. Travel agents are able to synergistically compose individual customized sales presentations and itineraries for their clients, representing thousands of tour destinations and criteria, from multiple permutations of data sources in a fully automated fashion." This patent is a continuation of three abandoned patents and presents CRSs as prior art. The patent mainly describes a means for composing and displaying individualized sales presentations according to determinants entered into the travel agent and based on a customer's profile and requests and in combination with remotely-accessible sources of information.

Comments

We recommend reading these patents (see the USPTO URL below).

Lockwood believes in videodisks and in audio-video experience.

We found later that users require a more interactive experience. Audio-video experience is time-consuming for both the user and the expensive self-service terminals. Furthermore the audio-video experience is very demanding in language skills: maybe 1 billion persons can effectively use an English site when only English speakers can appreciate an audio-video experience. Audio-video experience is also expensive to produce and culture dependent. We learnt that audio experience is not effective in a commercial location. Lockwood did not predict the evolution of the society, for instance the rise of delinquency and vandalism. Please do not see here a political statement. When you work on flight check-in software you get an instruction: "Everything that can be stolen will be stolen. Do not even use a mouse". A self-service terminal is armored today. Such features increase the device cost and reduce its application to short operations like getting money on an ATM, which is good for the device that can serve more users and for the user who stands less time in a vulnerable position.

Lockwood missed the most important revolutions ongoing in 1980 s:

1. Xanadu and other hypertext projects, which were at the origin of the Web,
2. PARC and Mac Intosh that changed the user interface,
3. Ubiquitous, inexpensive because mass produced personal computers,
4. Internet, a general purpose network where providers and consumers could meet.

The current environment is different of what was envisioned in Lockwood inventions. There is no single organization that control terminals, networks and central systems. Internet, where users and server systems connect, acts as a market place. Users own their equipment. Servers process requests and manage data.

Lockwood was right in 4,567,359 when he identified a customer need for getting insurance quotations and

Patent search

policies online. But we believe that he missed a key point that prevented the extension of the CRS concept to other industries: Travel industry has a large number of products that often complement rather than compete with each other. Sharing information, for instance schedules and fares, actually increased the total revenue of Travel industry. Insurance companies had no business case to create a centralized system because they had less products and more comprehensive offers.

Story

In 1996 Lockwood sued American Airlines (the CRS inventor) asserting that a new product of American Airlines allowing to access schedule and fare information, to book itineraries, and to retrieve photographs of places of interest, including hotels, restaurants, and cruises, for display to consumers infringed his three patents. In a first judgement the district court found that the American Airlines didn't meet the conditions of the Lockwood invention:

1. The Travel agent rather than the customer used the American product. The American product was a program and not a "substantially self-contained apparatus dimensioned to be easily transported". This product lacked "audio-visual means," "customer operated means," and "means for collecting payment."
2. The American product did not compose "individualized sales presentations" or "selectively combine" customer information to retrieve its photographs

The district court also found that

- 4,567,359 was obvious in light of the American CRS, SABRE and of 4,359,631/RE32,115,
- Claims of 5,309,355 were anticipated in 4,567,359.

Therefore the court invalidated 4,567,359 and 5,309,355.

Patent structure

To correctly read the judgement you must understand how patents read in court.

A patent contains:

- An abstract
- Claims
- A description itself divided in Field of the Invention, Background of the Invention, Summary and Description of the preferred embodiment

If you are just interested by learning about the IT system of your competitors, you can focus on the description of the preferred embodiment, usually the most readable part for programmers. But if you want to know if you infringe a patent you must focus on the claim section that defines the scope of the patent owner's exclusionary rights. The rest of the patent exists because:

Patent search

- The counterpart of the granted exclusionary rights is the publication of the invention details. After expiration of the patent, the public has free use of the invention and should be able to reconstruct it from the description. This is called enablement.
- When the meaning of a term used in a claim is not clear, an enhanced understanding may come from the description of the invention

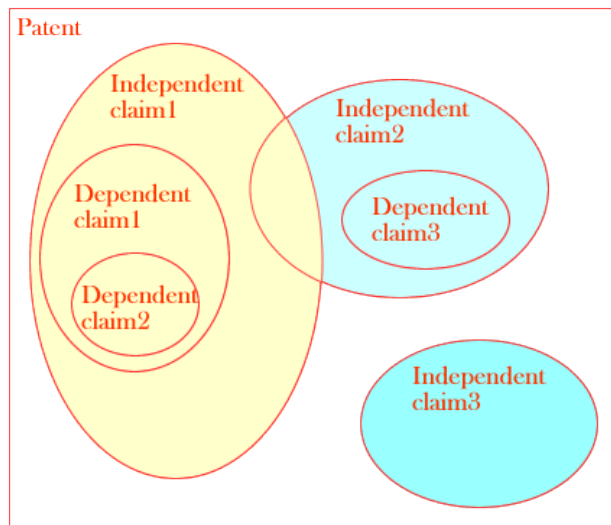
There are two sorts of claims:

- Dependent claims that refer to another claims. A dependent claim starts with "the system of claim ..." or in case of 5,309,355 with "The merchandising apparatus of claim 1..."
- Independent claims

Each claim defines a scope. Independent claims are the most important, they stand on their own and they provide a complete definition. It is enough to infringe an independent claim to infringe a patent. Dependent claims mainly exist to provide a fallback protection if the independent claim is invalidated.

Given an independent claim that describes three elements, element1, element2 and element3, you infringe the patent if your process uses element1 AND element2 AND element3.

Given a dependent claim n that depends on a dependent claim j that itself depends on an independent claim i, your process infringes claim n only if your process infringes the claim i AND claim j AND claim n.



Here is the way it works:

- The patent protection scope is the union of the claim arrays
- If you find prior art in a claim array then the claim is invalidated
- If your process sits in a valid claim array then you infringe a patent

Independent claims being by definition the broadest in scope are the most vulnerable.

Patent search

Actually you can also infringe the patent if the court finds that your process employs essentially the same method in essentially the same way to achieve essentially the same result as the invention. This is called the doctrine of equivalence. There was a restriction in USA between November 2000 and May 2002 to the doctrine of equivalence defined by the Festo ruling. On November 29, 2000, the United States Court of Appeals held; (1) Any reason for amendment to a patent claim that is related to patentability will give rise to prosecution history estoppel; and (2) When the amendment creates a prosecution history estoppel, there is no range of equivalents available for the amended elements

Just in case you do not know what estoppel means, here is the explanation. If someone states that something is so and, in reliance upon that statement, another person acts in a particular way, possibly to their detriment, then the person who made the statement is prevented, or estopped, from denying the correctness of the statement which they originally made. During the patent prosecution an examiner checks the claim novelty. When the examiner objects to a claim the applicant usually amends this claim. If the examiner finds no objection to the amended version of the claim the patent is granted. If the patentee (the former applicant) uses the doctrine of equivalence for the amended claim it denies in some way the correctness of the amendment that allowed the examiner to grant the patent.

On May 28, 2002 the Supreme Court vacated the Festo decision. As reported here the Supreme Court acknowledged that the doctrine of equivalents introduces uncertainty into the issue of claim interpretation, however, the Court disagreed with the complete bar rule set out by Federal Circuit, preferring instead a flexible approach to the doctrine of equivalents. While the Court agreed that any narrowing amendment made for a reason related to patentability could give rise to prosecution history estoppel, the Court stated that the Federal Circuit ignored the instruction in Warner–Jenkinson "that courts must be cautious before adopting changes that disrupt the settled expectations of the inventing community. Inventors who amended their claims under the previous case law had no reason to believe that they were conceding all equivalents" of amended elements when responding to a rejection. Festo decision being retroactive the Supreme Court was arguably true on this point. You can find the Supreme Court decision here.

The cancellation of the Festo decision does not mean that the history of the patent prosecution can be ignored. Before Festo "previous decisions had held that prosecution history estoppel constituted a flexible bar, foreclosing some, but not all, claims of equivalence, depending on the purpose of the amendment and the alterations in the text." Basically if the claim was narrowed because of prior art and in light of the nature of this prior art the doctrine of equivalence could not apply. This is what the Supreme Court calls the flexible bar. The consequence of this cancellation is that we are again in the same situation as before 2000. The Supreme Court wrote: "The doctrine of equivalents allows the patentee to claim those insubstantial alterations that were not captured in drafting the original patent claim but which could be created through trivial changes. When, however, the patentee originally claimed the subject matter alleged to infringe but then narrowed the claim in response to a rejection, he may not argue that the surrendered territory comprised unforeseen subject matter that should be deemed equivalent to the literal claims of the issued patent."

Therefore the Supreme Court concedes that a patent has to be interpreted in light of the proceedings in the PTO during the application process, which makes difficult for a patent holder to know what he owns, and for the public to know what he does not. This complexity makes of the patent system a tool for rich corporations. We must however say that the application of the doctrine of equivalence has never been easy, before like during and after Festo ruling. Before Festo applicants tend to file patents with broad claims that examiners had to narrow. With Festo ruling this method did not work anymore and applicants wrote more claims, narrow

Patent search

enough to not be rejected, which made patents harder to read.

Similar restrictions may exist in other countries. Reading the claims of a patent may be not enough to state if a product infringes a patent. You must study the correspondence between the applicant and the examiner. To get a first feeling you may look at the Patent Application Information Retrieval (PAIR) page of USPTO that we present below. Also carefully read claims. As explained on <http://www.iusmentis.com/patents/crashcourse/rights/> "everyone knows a nail is equivalent to a screw, and that the accepted general term is 'fastening means'. If a claim then nevertheless mentions the use of a nail to connect two elements, the chance that a court will rule the use of a screw to be equivalent is small. After all, when the patent does not use the general term, the author must have had a reason to use the specific term." Therefore when you find for instance the same specific name or adjective in all independent claims you may suspect that the claims were amended.

There is a nice explanation of scope at <http://www.iusmentis.com/patents/claims/>.

For more information look at <http://www.iusmentis.com/patents/crashcourse/procedure/>, http://www.baypatents.com/glossary/default.asp?ID_Glossary=27, <http://www.tms.org/pubs/journals/JOM/matters/matters-0102.html>, <http://inventors.about.com/library/weekly/aapatentdescriptionsa.htm>.

Story continued

Lockwood appealed. The decision of the appeal court is also worth some analysis.

First the court noted that though the American CRS, SABRE was introduced to the public in 1962 the essential algorithms of the SABRE software were proprietary and confidential and therefore not enough information had been made apparent to enable one skilled in the art to duplicate the system. Therefore the internal nature of the Sabre system could not be considered as prior art. But the court confirmed the decision of the district court by finding that the aspects of the SABRE system that made 5,309,355 obvious were public

The whole story is <http://www.law.emory.edu/fedcircuit/mar97/96-1168.html>. Note that some moments the decision refers to '631 (4,359,631) and at others '115 (for RE32,115, the renewal of 4,359,631).

Then Lockwood filed a patent 5,576,951, continuation-in-part of 4,567,359 and 5,309,355, describing an "Automated Sales and services system" and a patent 6,289,319 describing an "automatic business and financial transaction processing system". In 5,576,951 Lockwood thoroughly presents the CRSs in their 1994 form as the prior art and describe improvements such as end consumer access using a multimedia terminal.

A company called PanIP represents Lockwood interests and sues small Internet companies. Here is an example of lawsuit filing with the text and drawings of 5,576,951 and 6,289,319. One of the defendants, Tim Beere created a Web site <http://www.youmaybenext.com> where you can find this comment: " First, through the support of the public and the help of a private prior art search firm, we have found substantial evidence undermining PanIP's patents. In May we used this evidence to file requests with the U.S. Patent and Trademark Office ("PTO") to re-examine and invalidate both of them. At the same time we moved to stay PanIP's lawsuits, and in June the federal court in San Diego issued an order staying the cases to allow our

Patent search

re-examination requests to be considered. In July, we received notice that the PTO has accepted our request for re-examination and agrees that we have raised "a substantial new question of patentability" as to the claims of the first of PanIP's patents. Thus, this patent will now be re-examined in its entirety and we are confident that it will be invalidated. We anticipate a similar finding with respect to the second PanIP patent shortly."

You can find the re-examination procedure on <http://www.uspto.gov/web/offices/pac/mpep/documents/2200.htm>. This procedure is a part of the Manual of Patent Examining Procedure. This mail lists the main documents of interest. We used to recommend visiting the Panip site, which was a good site about Intellectual Property. Regarding the dispute above you can find decisions regarding a trademark infringement and a defamation by <http://www.youmaybenext.com>.

October 2004 update:

The defendants created a group called the PanIP Group Defense Fund, Inc. (PGDF).

On March 22, 2004, PanIP has dismissed its case against the PGDF and its individual members in exchange for a waiver of the attorney fee award granted to the PGDF. The PGDF did not take a license of any kind from PanIP.

The U.S. Patent and Trademark Office has issued an initial rejection of the claims in the re-examination of 5,576,951. On May 5, 2004, this news story said: "nine of the 10 individual claims in the patent were rejected on the basis of anticipation in prior art. That essentially means that the Patent Office examiner determined that every element that the patent holder claimed to originate actually predated the patent and could be found in a single reference, such as an article [...]. The remaining claim in the patent was rejected on the basis of obviousness, meaning that most elements of the claim existed in prior art and that the remaining elements would have been obvious to someone skilled in the art." I understand that this re-examination takes place with an application number 90/006,625.

The U.S. Patent and Trademark Office also granted the re-examination of 6,289,319. I understand that this re-examination takes place with an application number 90/006,623. I analyze the PanIP re-examination documents on the Examination page.

The PanIP entry page contained on October 3, 2004 a notice: "PanIP.com has been taken down for maintenance for maintenance and redesign, in view of new developments. Please return soon!"

PAIR

On November 3, 2003 we checked the patent status of 6,289,319 on the Patent Application Information Retrieval (PAIR) page of USPTO. We found this prosecution history:

#	Date	Content description
32	05-14-2001	Mail Notice of Allowance
31	05-14-2001	Notice of Allowance Data Verification Completed
30	05-14-2001	Notice of Allowability

Patent search

- 29 10-06-2000 Case Docketed to Examiner in GAU
- 28 09-27-2000 Mail BPAI Decision on Appeal – Reversed
- 27 09-27-2000 BPAI Decision – Examiner Reversed
- 26 10-25-1999 Mail Miscellaneous Communication to Applicant
- 25 10-25-1999 Miscellaneous Communication to Applicant – No Action Count
- 24 07-16-1999 Remand to the Examiner by BPAI
- 23 02-21-1997 Mail Reply Brief Noted by Examiner
- 22 02-21-1997 Reply Brief Noted by Examiner
- 21 12-26-1996 Date Forwarded to Examiner
- 20 12-18-1996 Reply Brief Filed
- 19 11-13-1996 Mail Examiner's Answer
- 18 10-28-1996 Examiner's Answer to Appeal Brief
- 17 07-17-1996 Date Forwarded to Examiner
- 16 07-12-1996 Appeal Brief Filed
- 15 07-12-1996 Request for Extension of Time – Granted
- 14 04-15-1996 Notice of Appeal Filed
- 13 03-27-1996 Mail Advisory Action (PTOL – 303)
- 12 03-26-1996 Advisory Action (PTOL-303)
- 11 03-13-1996 Date Forwarded to Examiner
- 10 03-11-1996 Amendment after Final Rejection
- 9 01-16-1996 Mail Final Rejection (PTOL – 326)
- 8 01-11-1996 Final Rejection
- 7 10-19-1995 Date Forwarded to Examiner
- 6 09-22-1995 Response after Non-Final Action
- 5 09-22-1995 Request for Extension of Time – Granted
- 4 04-20-1995 Mail Non-Final Rejection
- 3 04-03-1995 Non-Final Rejection
- 2 02-03-1995 Case Docketed to Examiner in GAU
- 1 01-26-1995 Application Captured on Microfilm

The application was filed on November 30, 1994 and granted on September 11, 2001, almost seven years later, much more than the average claimed by USPTO (24.6 months). The following events happened:

1. Non-final rejection
2. Final rejection
3. Amendment
4. Appeal brief
5. Appeal to the Board of Patent Appeals and Interferences (BPAI)

The two first events are common. The examiner rejects the application usually because she finds that a claim is obvious or not novel. He reports his findings to the applicant. In case of final rejection the examiner fills a PTOL-326 form. After a final rejection the applicant can abandon her application, use a Continuing

Patent search

Prosecution Application practice that permit applicants to obtain further examination of an application or appeal against the examiner decision, which was the choice of Lawrence B. Lockwood. An amendment consists mainly in claim changes. Some claims can be cancelled, some others may be added and some others may be modified. The brief is a list of the authorities and arguments on which the applicant will rely to maintain his or her appeal. The BPAI may have been involved because of a procedure problem. See [MPEP #2274](#) and the [Examination page](#) for details.

The patent exercise is not cheap at ten to twenty thousand dollars in USA, ten times more for a Worldwide application. This exercise is also time-consuming: two months for patent writing and many requests to reply during two to seven years. You also need the help of a patent attorney. We found this comment in a PTOL-326: "An examination of this application reveals that applicant is unfamiliar with patent prosecuting procedure. While an inventor may prosecute the application, lack of skill in this field usually acts as a liability in affording the maximum protection for the invention disclosed. Applicant is advised to secure the services of a registered patent attorney or agent to prosecute the application, since the value of a patent is largely dependent upon skillful preparation and prosecution."

Eolas

We would like to turn now our attention on another example that recently hit the headlines, the Eolas v. Microsoft case. This example illustrates several aspects of software patents like the re-examination procedure. The story is not ended but in any case this case can have implications of interest.

A first interesting aspect is that the suit was initially unnoticed. An [Internetnews document](#) of February 2, 1999 contained: "[Eolas Technologies Inc.](#), a Chicago-based Internet technology firm, filed suit for infringement of its patent [5,838,906] on browser technology, charging that Microsoft's Windows 98, Windows 95 and Internet Explorer programs infringe on Eolas' worldwide exclusive commercial rights to interactive programs." The main plaintiff was Michael D. Doyle. Some sources say that he is the only employee of Eolas. The University of California, which was the patent assignee helped him in the trial and maybe also in the patent application. The patent is not without merit. It cites an unusually large number of references and was well written. We may also note that Eolas sued Microsoft almost as soon as it was granted the patent (November 17, 1998).

The first claim of this patent filed in October 1994 contains: "a first distributed hypermedia document includes an embed text format, located at a first location in said first distributed hypermedia document, that specifies the location of at least a portion of an object external to the first distributed hypermedia document, wherein said object has type information associated with it utilized by said browser to identify and locate an executable application external to the first distributed hypermedia document, and wherein said embed text format is parsed by said browser to automatically invoke said executable application to execute on said client workstation in order to display said object and enable interactive processing of said object within a display area created at said first location within the portion of said first distributed hypermedia document being displayed in said first browser-controlled window." The patent claims just cover the way applets, plug-ins, scriptlets and Active X controls are working.

It may be useful to remind that Mosaic, the first workable browser started to be widely used on Unix sites in 1994 and Netscape came soon after. The first version of Java was released in 1995 and was designed

Patent search

primarily to be used in applets. In the confusion a patent could come unnoticed. However this is not what happened for 5,838,906. On November 6, 2003 we checked its status on the [Patent Application Information Retrieval \(PAIR\)](#) page of USPTO. We found this prosecution history:

#	Date	Content description
57	11-05-2003	Application scanned in CRU and can be accessed by the REPS system in the Public Search Room
56	10-30-2003	Record a Petition Decision of Granted for Commissioner-Initiated Re-examination Proceeding
55	10-30-2003	Petition Entered
...
38	11-07-1994	Information Disclosure Statement (IDS) Filed
37	03-30-1998	Mail Notice of Allowance
36	03-30-1998	Notice of Allowance Data Verification Completed
35	01-27-1998	Examiner Interview Summary Record (PTOL - 413)
34	03-30-1998	Notice of Allowability
33	03-02-1998	Notice of Appeal Filed
32	03-02-1998	Request for Extension of Time - Granted
31	01-22-1998	Date Forwarded to Examiner
30	12-29-1997	Amendment after Final Rejection
29	12-29-1997	Supplemental Papers - Oath or Declaration
28	12-29-1997	Request for Extension of Time - Granted
27	11-06-1997	Examiner Interview Summary Record (PTOL - 413)
26	11-06-1997	Examiner Interview Summary Record (PTOL - 413)
25	08-25-1997	Mail Final Rejection (PTOL - 326)
24	08-22-1997	Final Rejection
23	06-19-1997	Date Forwarded to Examiner
22	06-05-1997	Response after Non-Final Action
21	03-26-1997	Mail Non-Final Rejection
20	03-21-1997	Non-Final Rejection
19	02-24-1997	Examiner Interview Summary Record (PTOL - 413)
18	02-20-1997	Date Forwarded to Examiner
17	02-19-1997	Amendment after Final Rejection
16	01-24-1997	Mail Final Rejection (PTOL - 326)
15	01-23-1997	Final Rejection
14	01-08-1997	Date Forwarded to Examiner
13	01-08-1997	Amendment after Final Rejection
12	01-08-1997	Affidavit(s) (Rule 131 or 132) or Exhibit(s) Received
11	12-13-1996	Mail Final Rejection (PTOL - 326)
10	12-12-1996	Final Rejection
9	10-09-1996	Date Forwarded to Examiner

Patent search

- 8 08–09–1996 Response after Non–Final Action
- 7 05–06–1996 Mail Non–Final Rejection
- 6 05–03–1996 Non–Final Rejection
- 5 04–08–1996 Case Docketed to Examiner in GAU
- 4 04–08–1996 Case Docketed to Examiner in GAU
- 3 03–14–1995 Application Captured on Microfilm
- 2 12–30–1994 Application Is Now Complete
- 1 11–18–1994 Incomplete Application under Rule 53(b) – Filing Date Assigned

The application was complete at the end of 1994, examined and rejected a first time in 1996, rejected a second time in August 1997 before being eventually accepted in March 1998. In 1996 and even more in 1997 the value of applets was apparent. The examiner could not ignore the implications of the claims when he chose to grant the patent. Furthermore because the Web was considered as a technological revolution there were plenty of books, articles and pages describing the main contributions. Never an examiner had easier access to prior art. From a legal point of view 5,838,906 was serious but nobody took it seriously.

5,838,906 being useful there were two solutions for the defendant:

- Demonstrate that there is prior art
- Demonstrate that the invention is obvious in light of the prior art

Prior art is made of expired patents and of everything that has been published or presented before. In USA and in some other countries like Australia you can publish or present your invention and file a patent later as far as a grace period has not expired. The grace period is of one year in USA. In their search for prior art, Microsoft turned up Pei Wei, who, in May of 1993 demonstrated Viola, a browser that integrated an application in basically the same way that is claimed in the patent. But Pei Wei developed Viola at University of California where Michael Doyle and other filed their patent in October 1994, therefore after the expiration of the grace period if the Viola demonstration was found valid. This is interesting to note that Microsoft did not try to demonstrate the obviousness of the invention. They could have objected that (1) the original browser of Tim Berners–Lee displayed images on different windows (2) the key breakthrough had been to embed pictures on the same window as text (3) the obvious next step was to embed controls able to communicate with servers, these controls being already used notably with Microsoft Visual Basic.

On August 12 2003 a Chicago jury ordered Microsoft to pay \$521 million to the University of California and Eolas Technology for its past infringement of 5,838,906 in Internet Explorer. Microsoft said it would appeal the ruling in the U.S. Circuit Court of Appeals. "We believe the evidence will ultimately show that there was no infringement of any kind, and that the accused feature in our browser technology was developed by our own engineers based on pre–existing Microsoft technology," At the same time Microsoft said it would work hard to ensure there is "very little if any impact on our customers." Microsoft said in a statement. Michael Wallent, general manager of the Windows Client Platform at Microsoft explained: The company decided to make changes to avoid any accruing liability if it lost a planned appeal of the jury verdict and because there could be the risk of an injunction as the case proceeds.

Up to then only internetnews.com showed an interest for the story. On August 29 internetnews.com reported that W3C said Microsoft's changes to the IE browser may affect a large number of existing Web pages and

Patent search

urged the Web community to start considering and contributing to the range of technical options available if the patents are rigidly enforced. "Microsoft has been representing to the world that they have no choice but to remove technology from the browser and disrupt the Internet," Doyle answered in an interview on September 19. "And I want to make it very clear that that is not the case. Microsoft has had in its power the ability to settle this case, and to the extent that they're refusing to settle, it's their decision." On October 6 Microsoft announced steps to address Eolas Patent Ruling. Microsoft originally announced substantial changes to the Default Handling of ActiveX Controls by Internet Explorer:

- They affect the <object>, <applet> and <embed> elements
- By default when an Active X control is loaded the user must acknowledge a message box that displays: "Press OK to continue loading the content of this page". The reason for this message is that 5,838,906 describes a method for the browser to automatically execute a program. If it requires the user to acknowledge the Microsoft browser no longer automatically executes a program and therefore no longer infringes the patent.
- There are three solutions to avoid the message box. The first one is to use DHTML to dynamically insert the control stuff. The second solution is to define the <object>, <applet> or <embed> element without any attribute or <param> child that could specify a URI. The third solution is to add a new attribute NOEXTERNALDATA attribute to the <object> element. We understand that this also applies for <embed> and <applet>. NOEXTERNALDATA is remarkable: it declares that the control does not make network accesses and prevents the control from receiving URI data from page parameters. However technically nothing can prevent a control written in a native language from networking.

You may have to do something for applets and ActiveX controls and Flash, Media Player, QuickTime and RealOne handling. For Web sites that usually do not extensively use these facilities the problem is manageable. For browser-based professional applications the message box is not an option and changes are extensive.

In January 2004 the court entered its final judgment in the Eolas case and Microsoft intends to appeal. The court stayed implementation of the judgment, including implementation of an injunction, until the appeal has been heard and decided. Then Microsoft made a step back: "Given the present legal status as well as requests made by partners and customers, Microsoft will, for the time being, not move ahead with the modest steps it intended to take to modify Windows and Internet Explorer." This is a good illustration of the switching costs in the software industry. If Internet Explorer was a regular product, customers would move to a competing product but if Microsoft settled a licence agreement with the patentee. But Internet Explorer has a huge market share and the switching costs associated to a migration to another browser are higher than the changes proposed by Microsoft. So Microsoft can show that they can design around the patent and cope with their internal switching costs, to the opposite of their customers and partners and the public interest is to keep the ActiveX handling in Internet Explorer unchanged.

Now we can turn our attention to the last development of the story, the patent re-examination asked by W3C. 5,838,906 describes a way of summoning content located on a server other than the one serving the page in question. The concern for W3C, which manages the HTML standard is that <applet>, <object> and <embed> tags in HTML may fall under the wording of the Eolas patent. W3C standards like HTML were produced according to a patent policy whose goal is to assure that standards produced under this policy can be implemented on a Royalty-Free (RF) basis. The fact is that even without 5,838,906 the W3C was considering

Patent search

using another policy allowing producing standards using patented technology. A Patent Policy Working Group (PPWG) "acknowledged a central conflict to the standardization process: Companies that spend serious time and effort coming up with the technology behind the standards may be reluctant to simply give away the rights to what they consider their intellectual property." The proposal of the PPWG, called Reasonable And Non-Discriminatory (RAND), faced a serious opposition and the W3C had to step back. The chairman of the W3C advisory committee, Daniel Weitzner wrote "this does NOT mean that W3C has made final decision in favor of a RF-only policy, nor does it mean that we have made a final determination about the role RAND licensing will play. The final decision about W3C's patent policy will be made after the PPWG has developed a new proposal, the public has had another chance to comment, and the W3C membership has had it's chance to express its views formally to the Director".

We first cite another excellent paper on InternetNews:

"Even as the W3C is insisting prior art is readily available, many wonder why this was never uncovered during the Microsoft/Eolas case that has been before the courts since 1999. According to W3C's Weitzner, efforts to have the jury consider the prior art in the HTML standard was not allowed for procedural reasons: "It [the prior art] wasn't rejected for any reason that won't allow the patent office to reexamine it. It wasn't presented to the jury because of procedural issues," he insisted.

For W3C's Weitzner, the issue is one of maintaining Web standards. "We are not involved with the political machinations here. Our fundamental commitment is that Web standards should be royalty free. The enforcement of this patent is a direct attack on the ability to participate in the Web on a royalty-free basis. It goes beyond a damage award that one of our members have to pay or wrangling between competitors," he argued.

You can find a good presentation of the re-examination process at <http://www.baypatents.com/reexam.asp>. The procedure for patent re-examination is at http://www.uspto.gov/web/offices/pac/mpep/documents/2200_2201.htm.

The W3C did two things:

- It sent a citation of prior art to the USPTO on October 24. You can find on the PAIR page above a Decision of Granted for Commissioner-Initiated Re-examination Proceeding: the re-examination is ongoing (55 – 56).
- Tim Berners-Lee sent a letter to the Under Secretary of Commerce for Intellectual Property, James E. Rogan on October 28.

Whereas Microsoft chose to minimize the impact of the IE change the W3C writes in the citation: "Although Microsoft's proposed redesign, as we understand it, involves only a small portion of Internet Explorer, it would render Microsoft's browser incompatible with globally-accepted standards and impair the operation of millions of Web pages. The cost to the larger World Wide Web community of fixing the problems created by such a change to Internet Explorer is incalculable, but would likely require changes to millions of Web pages, as well as changes to Web page authoring tools and other software and systems designed for the World Wide Web."

Then the W3C writes: "We strongly believe that the 5,838,906 patent is invalid in view of prior art, submitted

Patent search

herewith, that was never previously considered by the United States Patent & Trademark Office. While we understand that the submitted prior art was introduced during the course of the recent trial proceedings, the issue of whether it renders the 5,838,906 patent invalid was never considered." Then the W3C recalls that embedding the image with text had been described before and already implemented in Microsoft Write. This is true but at the same time few inventions and almost no software invention stand if you accept this reasoning. Furthermore 5,838,906 cites the <embed> element and OLE. W3C understand this and presents two references published more than one year prior to the filing date of the 5,838,906 patent, which are messages posted on a mailing list and subsequently disseminated. Such documents have a status of printed document because "An electronic publication, including an on-line database or Internet publication, is considered to be a 'printed publication' within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates." You can find one of these documents at <http://ksi.cpsc.ucalgary.ca/archives/WWW-TALK/www-talk-1993q2.messages/467.html>.

W3C finds that claims 1-3 and 6-8 are obvious, which actually follows the understanding of Microsoft: They did not remove <embed>, <applet> and <object> from IE. They just disabled the capability for the called applets and controls to use page parameters to invoke a remote server, therefore precisely the function covered by claims 4-5 and 9-10 that stand, especially when read in light of the description: "However, this open distributed hypermedia system as it currently exists has shortcomings in that today's large data objects are limited largely by bandwidth constraints in the various communication links in the Internet and localized networks, and by the limited processing power, or computing constraints, of small computer systems normally provided to most users. Large data objects are difficult to update at frame rates fast enough (e.g., 30 frames per second) to achieve smooth animation. Moreover, the processing power needed to perform the calculations to animate such images in real time does not exist on most workstations, not to mention personal computers." This sentence lists reasons why programs like RealOne were developed and browser applications were so appealing for professional users.

The second document is the letter of Tim Berners-Lee. In this letter he stresses the disruptive effect of the court decision. For us the strongest argument is the "impact on millions of Web pages that may have historical importance". But we are not entirely convinced by the reasoning. A historical page cannot depend on a proprietary and unpublished technology. Historical pages are usually written in pure HTML because proprietary technologies require the use of proprietary authoring tools and imply higher authoring costs than HTML. The court could also find that

1. the plug-ins (including the free Java and Flash plug-ins) versions are frequently incompatible and cannot be used for historical content,
2. internet archives (like archive.org) don't properly handle non-html content

Tim Berners-Lee writes: "The impact of the '906 patent reaches far beyond a single vendor and even beyond those who could be alleged to infringe the patent. The existence of the patent and associated licensing demands compels many developers of Web browsers, Web pages, and many other important components of the Web to deviate from the fundamental technical standards that enable the Web to function as a coherent system. In many cases, those who will be forced to incur the cost of modifying Web pages or software applications do not even themselves infringe the patent (assuming it is even valid). Given the interdependence of Web technology, those who wrote Web pages or developed software in reliance on Web standards will now have to retrofit their systems in order to accommodate deviations from standards forced by the '906 patent.

Patent search

These deviations will either reflect individual decisions by developers about how to avoid infringement liability, or will be an effort to be compatible with decisions individual vendors make in the course of their own re-design. What's more, the inevitable fragmentation and re-tooling costs caused by the ability to enforce this patent, which we believe to be invalid, cannot even be remedied by individual parties choosing simply to pay licensing fees to the patent holder. If some parties are granted a license, while others either don't or can't obtain one, we will still be left with impaired functionality of the Web. Global standards have been the basis of assuring interoperability on the Web. A patent whose validity is demonstrably in doubt ought not be allowed to undo the years of work that have gone into building the Web."

"Removing the improperly disruptive effect of this invalid patent is important not only for the future of the Web, but also for the past. Even if the Web has to endure several years of disruption, we are confident that currently active Web pages will eventually be fixed and brought into compliance with whatever the prevailing standard is. However, pages that are inactive but have historical value may well remain in a state of impaired accessibility indefinitely if Web technology is forced to deviate from standards in this manner."

This case demonstrates that the software patent law itself yields surprising and unexpected results. Thank to a succession of errors the industry used a patented technology to implement key functions. The cost of bringing in compliance pages and applications may be of the same order of magnitude as the so-called 2000 bug. A decision consistent with the current practice would be the invalidation of the 5,838,906 claims 1-3 and 6-8, which should be sufficient for historical pages. This decision can popularize and facilitate re-examination. A possible consequence could be that each time a company A sues company B for patent infringement company B asks for the re-examination of the patent. 5,838,906 does not demonstrate that software patents are bad. First this case could demonstrate that a single person can win against industry interests. Second this case may help improving the examination process.

October 2004 update:

The re-examination of 5,838,906 is ongoing. I understand that this re-examination takes place with an application number 90/006,831. On February 26, 2004 and August 8, 2004 two non-final actions rejected the claims of 5,838,906. I analyze the Eolas re-examination documents on the [Examination page](#).

Patent organization

Patent lifecycle

The process for getting a patent is:

1. The inventors file a patent application at a national patent office.
2. The patent office makes a novelty search.
3. An Examiner examines the patentability of the invention in light of the report of the novelty search. An invention is not novel if all the elements that are mentioned in a claim are also found in a single document that is part of the prior art. Any document from any source can in fact be used as prior art. It does not matter in which language the document is written, in how many copies it was made available, or whether any copies were in fact bought or read by third parties. The key point is that the

Patent search

document had to be published before the date of filing. As a consequence a web page can only be considered as prior art if it can be proven that this web page was publicly accessible the day before the filing date and with its content the day before the filing date.

4. The inventors and their attorney can amend the application if it was found obvious or not novel.
5. The application is published 18 months after it has been filed. It is still an application and can still be rejected. Very often the inventors add a statement "patent pending" on their site when the application is not yet published to deter competitors to develop similar products.
6. The examiner issues a notice of allowance and the patent is granted to inventors.

The granted patent protects the invention only in the application country. As we seen in the example above the patent can still be invalidated by a court as it was the case for the first PanIP patent. In USA it can also be re-examined by the patent office as it is currently the case for the Eolas patent and for other PanIP patents. For more information on the examination and re-examination you can read the [re-examination page](#).

Patent cooperation treaty

To get patent protection in other countries the inventor file one single patent application to a World Intellectual Property Organization (WIPO) office in which it designates the countries in which he wants to have patent protection. Thank to the Patent Cooperation Treaty (PCT) the patent application will be applied in parallel in these countries with the filing date of the original application country.

Somewhat the same process as above is applied again:

1. An international bureau makes a search for prior art
2. National patent office, which can also make searches for prior art, examine the patent

This promotion process implies that governments agree on a common understanding of patents. This is defined by an agreement signed at the end of the Uruguay round, the [Agreement on Trade-Related aspects of Intellectual Property rights](#) (TRIPs). This document leaves some space for interpretation. You can look at a [Review of the implementing legislation](#) for government answers to questions asked by other governments. The fact is that software patentability is one of the most debated questions. You can download answers in a WordPerfect format that you can read with Word.

Patent offices

Country	Site
Australia	The home page is http://www.ipaustralia.gov.au/index.shtml . You can find the search engines of Intellectual Property Australia at http://pericles.ipaustralia.gov.au/ols/searching/content/olsPatents.jsp . The easiest to use search engine is http://apa.hpa.com.au:8080/ipapa/asearch
Canada	http://strategis.ic.gc.ca/sc_mrksv/cipo/patents/pt_main-e.html contains interesting documents. For searches use http://patents1.ic.gc.ca/srch_adv-e.html .
Europe	http://www.european-patent-office.org/index.htm and http://www.european-patent-office.org

Patent search

are two URLs of interest. The country sites are on <http://www.european-patent-office.org/espacenet/info/access.htm>. For searches use <http://ep.espacenet.com/>. The EPO Online file inspection is the equivalent of the USPTO s PAIR.

Japan	For searches use http://www.ipdl.jpo.go.jp/homepg_e.ipdl .
New Zealand	The Intellectual Property site is http://www.iponz.govt.nz/pls/web/dbssiten.main .
Singapore	The Intellectual Property site of Singapore (IPOS) is http://www.ipos.gov.sg/ . For searches use http://www.epatents.gov.sg/PE/ and http://www.surfip.gov.sg/sip/site/sip_home.htm . The former URL is the URL of the search engine of the Singapore Intellectual Property database. The latter URL is the URL of a portal allowing searching the Japanese and Taiwanese Intellectual Property databases too.
USA	The Intellectual Property site is http://www.uspto.gov/ . For searches use http://www.uspto.gov/patft/index.html . To inspect the correspondence between the applicant and the USPTO use http://portal.uspto.gov/external/portal/pair .
Word-wide	The URL is http://www.wipo.int/ . For searches use http://ipdl.wipo.int/ .

The European Patent Office (EPO) is a prerequisite to European Union adhesion. However the EPO does not depend on European institutions (Commission and Parliament). The EPO site holds data on patents from 71 countries (45 millions at the beginning of 2004) and applications published by WIPO. Since 2003 their Online file inspection allows reading and downloading the correspondence between the between applicants and examiners thank to an Article 128 of the European Patent Convention, according to which the public is entitled to inspect the complete contents of the files relating to all European patent applications after they have been published. Since September 2004 the USPTO s Patent Application Information Retrieval (PAIR) implements the same function. The USPTO hold data on US patents since 1790 and US applications published since 2001.

You may look at <http://www.piug.org/patdbase.html> for a more comprehensive list of patent databases and at <http://www.piug.org/patoffc.html> for a fairly comprehensive list of patent offices.

Patent search

Patent search is almost like Web search. The differences are:

1. All patents and only patents are available. In that respect patent search engines compare more to search engines like htdig setup to index a given site than to public search engines like Google, which only indexes the most popular pages regardless of their nature. The most competitive words of a patent search engine are not the same as the most competitive words of a public search engine. If you look for seat and computer with a public search engine you will get inconsistent results whereas if you make the same search on a Patent search engine you will mainly find transportation industry patents. On the other hand acronyms, which are effective with Web search engine thank to the popularity rule do not work with patent search engines
2. A patent is a structured content, therefore you can search for instance on claims or abstract
3. Patents are classified and you search for patents in a certain class.

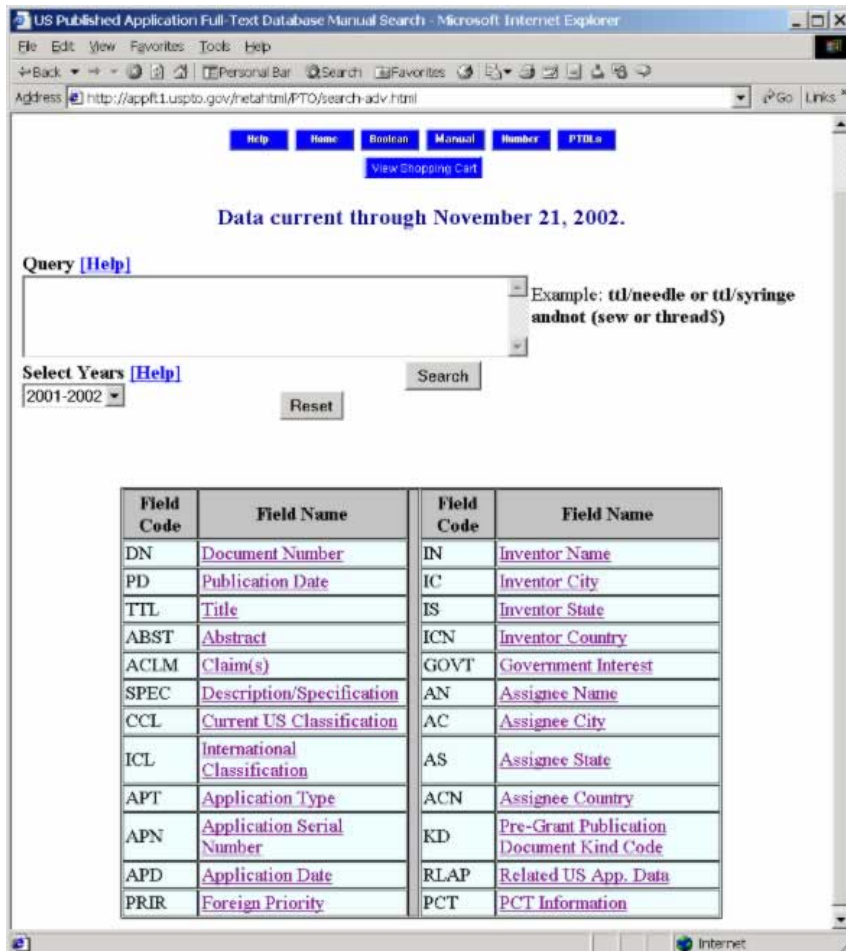
Patent search

4. A public search engine like Google indexes 3 billions patents. There are millions and not billions patents.

In this section we will take USPTO as an example.

USPTO search

The patent search page is <http://www.uspto.gov/patft/index.html>. Select Advanced search in Published Applications. You should get this:



Let's assume we want to find patent applications related to PageBox. PageBox is a computer program that allows deploying applications to remote locations. The best keywords are computer, deployment, deploy and remote. If we were using a Web search engine, we would rather use Web service, deployment, J2EE, Java.

Then we will restrict our research to the text of the claims. The query will be:

aclm/computer and aclm/remote and aclm/deploy\$

Patent search

We get a list of 46 patent applications on Nov 26, 2002. 20020129129 (System and method for deploying and implementing software applications over a distributed network) is pretty close in some respect to PageBox. 20020129129 has been classified as 709/220, 709/218 by USPTO and G06F 015/16; G06F 015/177. We can think that if PageBox was patented it would also be in this classes:

709 stands for "MULTIPLE COMPUTER OR PROCESS COORDINATING", 220 for "Network computer configuring" and 218 for "Using interconnected networks." G06F 015/16 stands for "Combinations of two or more digital computers each having at least an arithmetic unit, a programme unit and a register, e.g. for a simultaneous processing of several programmes" and G06F 015/177 for "Initialisation or configuration control."

Computer often but not necessarily exists in the claims of a PageBox related patent but we found that such a patent should be in 709/220. We improve our result with:

acm/remote and acm/deploy\$ and ccl/709/220

When a competitor claims to have a patent pending we use that sort of fast search. To identify patents that a project or product we use different method that consists:

1. In writing a reference text or at least a definition
2. Analyzing the reference text to extract the keywords

Our advice is to forget common sense. A patent is a text made of sentences. The most effective solution is therefore to find texts that look like a reference text. It is basically the same concept but not the same process as the NSA patent 5,937,422, "Automatically generating a topic description for text and searching and sorting text by topic using the same."

Note:

In the example above we used "Advanced search in Published Applications" that allows displaying patent applications, which are no longer secret and not yet granted. On <http://www.uspto.gov/patft/index.html> you can also select "Advanced search in Issued Patents" to display granted patents.

Which company filed the application?

In case of 20020129129 and 5,937,422 the assignee name is the name of the company. However, quite often, the assignee name is the name of the patent attorney.

If you suspect that the assignee name is not the company, look:

1. for the patent in the EPO site, and if you don't find the application,
2. at the inventor names,
3. at the specification examples and drawings,
4. at the inventor addresses.

Patent search

EPO

To look for US patents on the EPO site this is simple: add a us prefix and remove the commas. To look for 5,937,422 enter us5937422. For applications you may need to make two trials. 20020129129 is actually 2002/0129129. So you must try with and without the zero after the slash, so us20020129129 or us2002129129 (in this case us2002129129 is correct.)

Inventor name

If the "inventor" is an Intellectual Property specialist you are out of luck.

Our experience is that there are three sorts of inventors in case of company application:

1. Inventors who designed the patented invention
2. People who contributed to the invention for instance by checking the timecards and the expense reports of the first category
3. VP, C*O who motivated the first category. They are quoted in press releases.

Our experience is that there are three types of inventors who design inventions:

1. The living treasure. The living treasure has a big office. He represents his company in technical conferences.
2. The maverick. The maverick does not like his job. He has a web site with his resume.
3. The modest person. Because modest persons do not even know that they invent, we collectively call them dark matter: we postulate that they exist because something was invented.

This is easy to identify VP, C*O, living treasures and mavericks with Web search engines. Except for living treasures who like stability you need to find out where they were working at the date the application was filed. The living treasure or the maverick are often at the beginning of the inventor list and the VP/C*O are often at the end of the inventor list.

Examples and drawings

If the examples chosen in the detailed description of the preferred embodiment always relate to the same company there is a good chance that this company is the applicant. Look also at the figures. Quite frequently for business method and software application there are a couple of screenshots with the logo and the name of the company.

Inventor address

The city and the state where the inventors live are provided. With a site like MapQuest you can easily figure how distant are the inventors and which city is at a commuting distance from inventors. Usually geographical data are not sufficient to be sure that a patent was filed by a company for the reason that in USA companies

working in a given sector are usually located in the same area.

Toolkit

Text analysis

Word count statistics give good result in patent search.

The principle is to measure the word frequency in a reference text and to use the most frequently used words in the search. Related patents are likely to exhibit similar word frequencies. The results are better when the reference text is of the same type as the searched section: use most frequent words in reference claims to search on claims and use most frequent words in a reference description to search on description.

If the reference text is written in Word you can use the WordCount macro of Cuckoo. This macro writes a report document like this:

91 2% patent
33 1% search patents
25 0% http
24 0% was ://
23 0% invention
22 0% process
20 0% claim
18 0% use claims www system
17 0% same
16 0% section application
15 0% art also
14 0% software 359
13 0% prior most lockwood text
12 0% exist company information only office travel web

Patent search

11 0% business find American when because

10 0% page filed from uspto inventor experience court

...

This macro was designed primarily to improve the wording and the page performance on public search engines. Regarding the latter aspect you can find other tools on the Web. The goal is to facilitate the work of the search engines robots. These robots use word frequencies in the displayed text to determine the page subject, the idea being that a user who makes a search with the page s most frequent words should find the page interesting.

We wrote a standalone tool to measure word frequencies in html pages also called WordCount.

WordCount

WordCount is written in Java (JDK 1.3 or 1.4) and uses the Kizna HTML parser.

HTMLParser is a library released under LGPL, which allows you to parse HTML (HTML 4.0 supported.) You can download HTMLParser from <http://htmlparser.sourceforge.net>.

Download

Source [WordCount.java](#)

Binary [WordCount.class](#)

Example of exclusion file [excl.txt](#)

WordCount is a command line tool. Assuming that you have included the JDK on your path and the WordCount.class directory and htmlparser.jar on your CLASSPATH you can measure the word frequencies in the 20020129129 patent application with:

```
java WordCount
```

```
"http://appft1.uspto.gov/netacgi/nph-Parser?
```

```
Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50
```

```
&s1='20020129129'.PGNR.&OS=DN/20020129129&RS=DN/20020129129" excl.txt
```

Where the first parameter is the URL of the patent application on the USPTO site and excl.txt is a list of words to exclude from the measurement. We should get a report like this:

```
Found 5783 words
```

```
2% 119 component$
```

Patent search

1% 91 client\$

1% 88 server\$

1% 86 handler\$

1% 82 device\$

1% 81 fil\$

1% 71 download\$

1% 70 system\$

1% 68 network\$

1% 58 document\$

0% 57 install\$

0% 55 element\$ script\$

0% 53 contain\$

0% 51 method\$

0% 49 application\$

0% 47 logic computer\$

0% 45 object\$

0% 41 operat\$

0% 39 window\$

0% 38 browser\$

0% 37 frame\$

0% 35 internet

0% 34 platform\$

0% 33 function\$

Patent search

0% 32 provide\$
0% 31 require\$ factory deploy\$
0% 30 invention
0% 29 database
0% 27 render\$ specific interface\$
0% 26 embedd\$ implement\$ schema
0% 25 process\$ button\$
0% 24 operation\$
0% 23 version\$ enabl\$
0% 22 visual assembler\$ installation manager wireles\$
0% 21 standard\$ invoke\$ us\$ layout
0% 20 support\$ present\$ event\$ starter task\$
...

The report contains one line per word frequency that contains:

- The word percentage
- The word count
- The words whose word count occurrences were found. In some cases the word is used with different suffixes. In this case only the beginning of the word is displayed followed by a \$.

WordCount analyzes the text in the HTML elements. It excludes the words in excl.txt and applies a few English-dependent rules:

1. It eliminates numbers
2. If a word differs of another word only by an ending s, d, ed, ing, both words are considered to be the same (see the ignore method in the source code)

Therefore for other languages you probably need to slightly modify the code.

We do not believe that an improved WordCount would be much more effective. Once you have generated the list you must manually select the unexpected frequent words to be the keywords of your queries.

Search revisited

Advanced search tools in Patent search engines and in Web search engines share a common drawback: they do not allow finding ALL documents of interest for you. It is usually a minor drawback in case of Web search but in case of patent search you need to be reasonably sure that your process infringes no patent.

With the help of WordCount we can build an unordered keyword vector, for instance in case of 20020129129 $V = \{ \text{"server"}, \text{"client"}, \text{"download"}, \text{"deploy$"}, \text{"remote"} \}$

Then we can define a patent distance dV to the reference text, using the vector V as

- 0 when the patent contains all keywords in V
- 1 when the patent contains all keywords in V except one
- 2 when the patent contains all keywords in V except two and so on

Then we can look at the patents at a given distance or closer. The idea is to recognize that there is some uncertainty in the wording of patents. We can expect that related patents use most of the words in the keyword bag but not necessarily all of them.

We can compare the lists found with this distance and with Boolean searches:

- If we make a search with server AND client AND download AND deploy\$ AND remote we find all patents at a distance of 0.
- If we make a search with server OR client OR download OR deploy\$ OR remote we find all patents at a distance of 4 or closer ($\text{size}(V)^{\text{TM}}$).
- We cannot find patents at a distance of 2 or closer.

It would be nice if search engines could recognize a syntax like this:

```
["server", "client", "download", "deploy$", "remote" ]2
```

to mean that we are looking at all patents at a distance of 2 (that contain at least three words in { "server", "client", "download", "deploy\$", "remote" }.)

This syntax is not supported. Therefore we need to write our own tools:

- Uspto processes queries to USPTO
- Intersect compares the responses returned by Uspto queries

Here is an example:

```
java Uspto uspto.txt "acm/server and ccl/709/220 and (acm/client or acm/deploy$ or acm/remote)" server
```

```
java Uspto uspto.txt "acm/client and ccl/709/220 and (acm/server or acm/deploy$ or acm/remote)" client
```

```
java Uspto uspto.txt "acm/deploy$ and ccl/709/220 and (acm/server or acm/client or acm/remote)" deploy
```

Patent search

```
java Uspto uspto.txt "aclm/remote and ccl/709/220 and (aclm/client or aclm/deploy$ or aclm/server)" remote
```

```
java Intersect intersect.html server20021130.txt client20021130.txt deploy20021130.txt remote20021130.txt
```

At each Uspto call we set one keyword as mandatory and the other keywords in an OR statement.

Then Intersect combines the Uspto reports to display a list of patents sorted by growing distance (patents that contain all keywords, then patents that contain all keywords except one and so on.)

Uspto

Uspto is written in Java (JDK 1.3 or 1.4.)

Download

Source [Uspto.java](#)

Binary [Uspto.class, Uspto\\$TextUrl.class](#)

Example of keyword file [uspto.txt](#)

Uspto is a command line tool whose syntax is:

```
Uspto keyword_file advanced_query_string report_file
```

Assuming that you have included the JDK on your path and the Uspto.class directory on your CLASSPATH you can list patents whose class is 709/220 and whose claims contain "server" and either "client" or "deploy" or "remote" with:

```
java Uspto uspto.txt "aclm/server and ccl/709/220 and (aclm/client or aclm/deploy$ or aclm/remote)" server
```

Here is an example of keyword file:

publish 2

subscrib 2

network 2

install 2

method 2

assembl 2

server 2 client

Patent search

Uspto calls the USPTO site in HTTP using the query parameter. Then Uspto parses the response and identifies the patent titles and URLs. Uspto calls again the USPTO site if the patent list cannot fit on one page. Next Uspto computes the patent title weights using the keyword file data.

A line in the keyword file contains two or more fields:

- If there are only two fields then when it finds this field in the patent Uspto adds the number in the second field to the patent weight
- If there are more than two fields then Uspto adds the number in the second field to the patent weight when the title contains the first field AND at least one other field (third field and next)

The weight idea is that we only read patents whose title sounds interesting and that it is boring to read 50 and more patent titles. Weights aim to facilitate this first selection. When you write a keyword file keep in mind that a title contains between two and twenty and more words. You cannot be specific.

Uspto creates two files:

- A HTML report named `report_file + timestamp + ".html"`, for instance [server20021130.html](#) where Uspto displays patents with a weight > 0
- A text report named `report_file + timestamp + ".txt"`, for instance [server20021130.txt](#) where Uspto stores the patent application number, weight, title and URL

Intersect

Intersect is written in Java (JDK 1.3 or 1.4.)

Download

Source [Intersect.java](#)

Binary [Intersect.class](#), [Intersect\\$AppliData.class](#)

Intersect is a command line tool whose syntax is:

`Intersect report_file uspto_report [uspto_report]*`

Assuming that you have included the JDK on your path and the `Intersect.class` directory on your CLASSPATH you can compare reports issued by Uspto on November 30, 2002 with:

```
java Intersect intersect.html server20021130.txt client20021130.txt deploy20021130.txt remote20021130.txt
```

Here is a [report](#) returned with the command listed in the "Search revisited" section. The report contains four columns:

- Application number
- Count = the number of query in which the patent was found. It is the opposite of the distance. With

Patent search

four Uspto reports a count of four means that the distance is zero, a count of three means that the distance is one and so on

- Weight: same meaning as on the Uspto report
- Summary: patent title

You can click on the application number to display the patent. However the link is a string like "http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF

&u=/netahtml/PTO/searchadv.html&r=15&p=1&f=G&l=50&d=PG01

&S1=((server.CLM.+AND+709/220.CCLS.)+AND+((client.CLM.+OR+deploy\$.CLM.)+OR+remote.CLM.))

&OS=aclm/server+and+ccl/709/220+and+(aclm/client+or+aclm/deploy\$+or+aclm/remote)

&RS=((ACLM/server+AND+CCL/709/220)+AND+((ACLM/client+OR+ACLM/deploy\$)+OR+ACLM/remote))"

which means that it can interpreted in a different way by the USPTO dynamic search engine the day after.

Our experience is that these tools help finding patents of interest. Because they are simple you can adapt them easily to your needs. Then you can run them from time to time to check new patent applications. The key issue when we use or modify these tools is to keep a broad-enough spectrum to be reasonably sure that all patents of interest are in it and small enough to be able to analyze the patents.

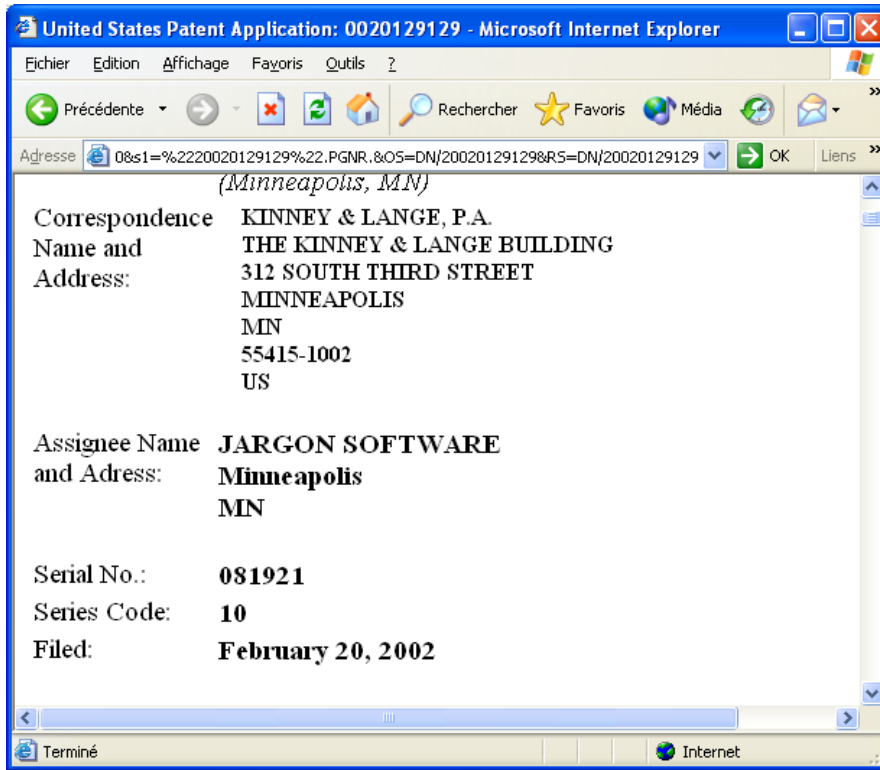
Prosecution analysis

USPTO

You may want to see the prosecution history of a patent (like 5,937,422) or an application 20020129129. For patents things are simple, just go on the PAIR (<http://portal.uspto.gov/external/portal/pair>), select Patent number in the Search method dropdown list and enter the patent number. For applications you must first find the correct application number.

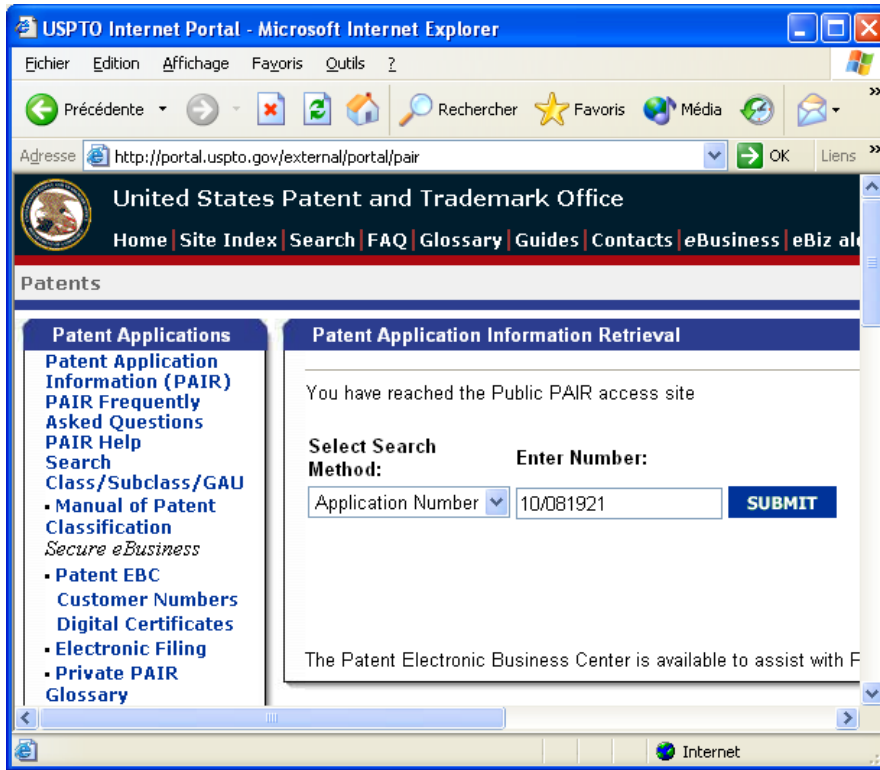
First go on <http://www.uspto.gov/> and search application by number. In the application file you should have something like this:

Patent search



The correct application number is made of the Series Code followed by / and by the Serial No. In this case this is 10/081921. Now you can go on <http://portal.uspto.gov/external/portal/pair> and enter this number:

Patent search



When you click on the Submit button you should get something like this:

Patent search

The screenshot shows a web browser window titled "USPTO Internet Portal - Microsoft Internet Explorer". The address bar contains a long URL. The main content area displays the following information:

10/081,921 System and method for deploying and implementing software applications over a distributed network

Navigation tabs: Application Data, Transaction History, Continuity Data, Published Documents, Publication Dates.

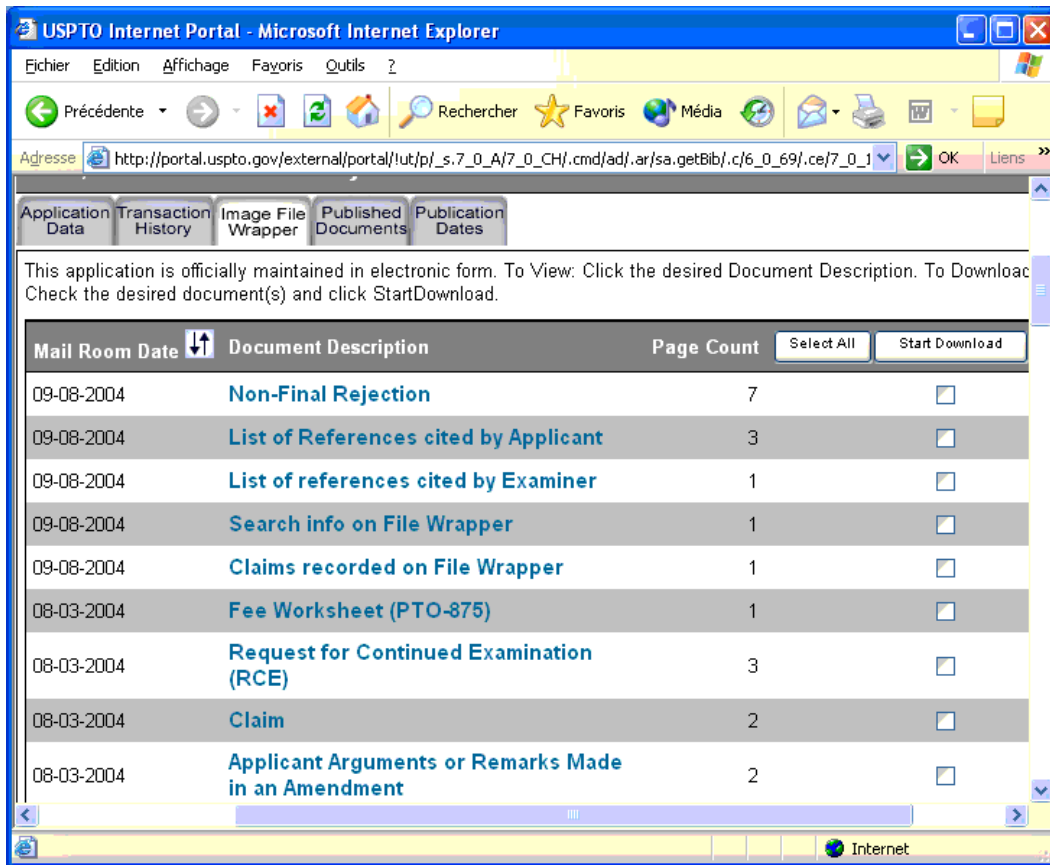
Application Number:	10/081,921	Customer Number:	-
Filing or 371 (c) Date:	02-20-2002	Status:	Docketed New Case - Ready for Examination
Application Type:	Utility	Status Date:	02-12-2003
Examiner Name:	HARVEY, JACK B	Location:	2100 CENTRAL FILES OVERI FRANCONIA
Group Art Unit:	2142	Location Date:	10-15-2003
Confirmation Number:	2203	Earliest Publication No:	US 2002-0129129 A1
Attorney Docket Number:	J267.12-0001	Earliest Publication Date:	09-12-2002
Class / Subclass:	709/218	Patent Number:	-
First Named Inventor:	Timothy Bloch , Sarasota, FL (US)	Issue Date of Patent:	-
Title of Invention:	System and method for deploying and implementing software applications over a distributed network		

At the bottom, there is a note: "The Patent Electronic Business Center is available to assist with PAIR questions at (866) 217.9197 (toll free) or EBC@user..."

In this case you have five tabs. The transaction history contains what we called so far the prosecution history. The continuity data are quite useful to list the other applications in the same family. Published data usually points the patent text we have seen above. Because the examination has not yet started in this case (status = Ready for Examination) the image tab is missing. This is the case also when the patent has been granted.

This image tab is present for the previous application (10/081921):

Patent search

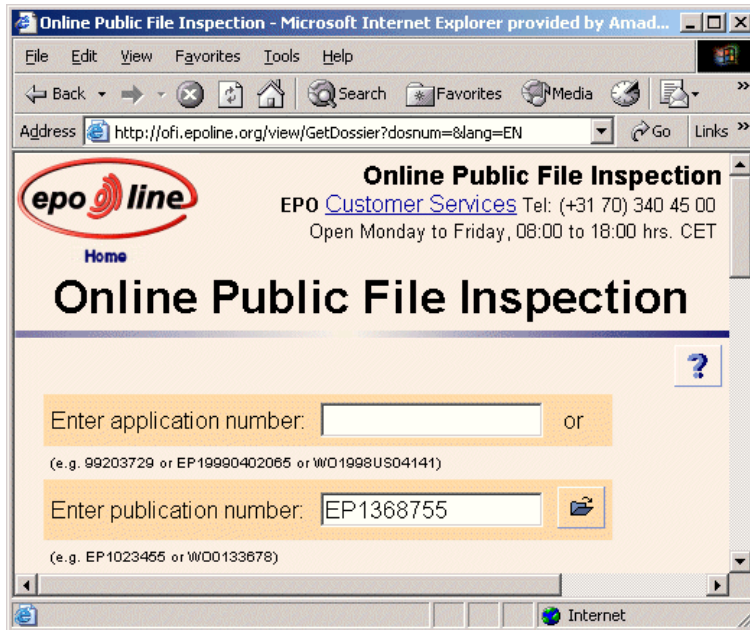


Click on a checkbox and then on the Start Download button to download the corresponding file in PDF format. It looks like the USPTO borrowed this not-so-obvious modus operandi from the EPO. Note that "Non-Final Rejection", "Applicant Arguments or Remarks Made in an Amendment" and the "Claim" coming with "Applicant Arguments or Remarks Made in an Amendment" are of interest:

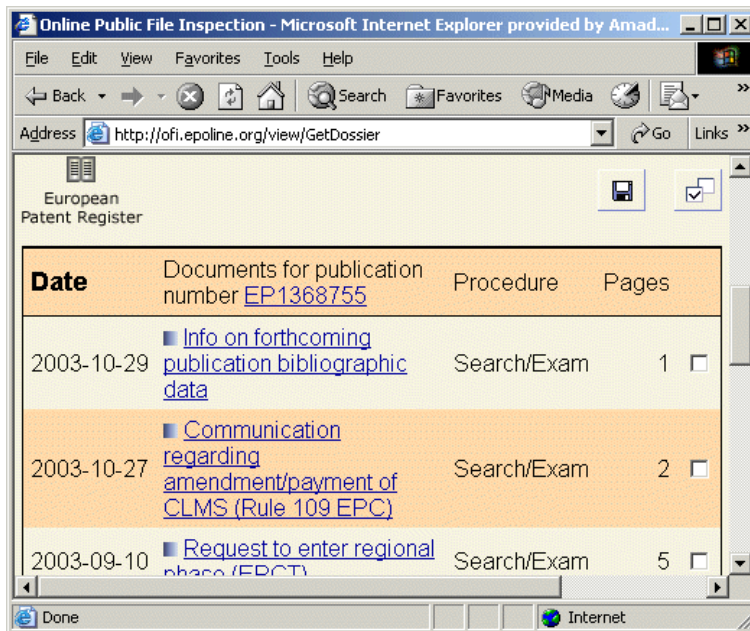
- In a "Non-Final Rejection" the examiner explains why he rejected the application.
- In an "Applicant Arguments or Remarks Made in an Amendment" the applicant explains why its application is patentable with the amended claims in the "Claim".

EPO

The EPO site contains prosecution files for applications and patents whose prosecution took place in Europe. To download these files go to <http://ofi.epoline.org/view/GetDossier> and enter a European application number, for instance EP1368755 (Method and system for providing message publishing on a dynamic page builder on the internet from IBM). You should have this:



Click on the Open this application icon. You should get this:



Click on a checkbox and then on the Start Download button to download the corresponding file in PDF format. Of a particular interest are:

- the international search report and examination (under PCT),
- the priority document (the original application, for instance a provisional application for an application first filed in the USPTO),
- the examinations,

- the amendments.

Open Source

As you can see PageBox is reasonably distant from patents covering similar processes. Furthermore the development started at the end of 1999, which means that there is prior art to oppose to an attempt of patenting a similar process. Normally this happens silently. The patent examiner checks that claims are novel. For this checking the examiner uses patent databases and Internet archives like archive.org (Internet Way back machine) where many PageBox pages are recorded. If she finds that a claim is not novel the examiner rejects the patent application.

Open Source developers do not necessarily check if their method or process is patented. This is understandable because the patentee usually sues the richest parties between the publisher and the product users.

For instance PageBox is released with an LGPL license that includes a no warranty clause: "BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION."

Therefore a company that would believe that PageBox infringes its patent would sue PageBox users rather than us and those users would have no means to complain to us. Large firms frequently perceive this aspect as a significant drawback of Open source. We must however clarify this issue.

The contract of a commercial program typically has a warranty and liabilities article that contains a clause like this: "The supplier must indemnify and hold harmless the customer against any claim concerning intellectual property relating to the program, provided that:

- The customer informs in writing the supplier of such claim in due time
- The customer assists the supplier and provides it with the information necessary for the enforcement of the above provision
- The supplier has the exclusive control of the defence and the entire freedom to sign a possible transaction."

The buyer of a commercial product may be involved in a patent dispute and if it does not settle a transaction with the patentee the publisher may just reimburse the paid fees to the buyer and not the training and switching expenses. These expenses are usually higher for a commercial product than for an Open Source product.

Patent issues

Enablement

In an Empirical Look at Software Patents Bessen and Hunt cite an IBM attorney: "[The patent standard] currently being applied in the U.S. invites the patenting of ideas that may have been visualized as desirable but have no foundation in terms of the research or development that may be required to enable their implementation." Here there are two observations easy to verify. The first one is that the patent purpose is to protect the function rather than the means. The second one is that inventors do not provide detailed instructions explaining how the invention works.

The second observation has serious implications. Patents were invented (1) to provide an incentive for inventing (2) to not lose knowledge. Before the introduction of patents inventors kept their inventions secret. As a consequence after one or two generation the invention was lost. Countries grant inventors "the right to exclude others from making, using, offering for sale, or selling" inventions in compensation of the invention disclosure. An invention whose description does not allow a person of the art to make the same product breaks this contract. Furthermore

- either the way to implement the invention is obvious, in which case either the invention itself is obvious, and hence not patentable or it is a Columbus egg; and is it possible to grant patents for Columbus eggs when there are one million specialists in any domain sufficiently important to motivate a patent filing and when disclosures are easy and cheap thank to the Web, the forums and the mailing lists? every Columbus egg has been invented (for instance at coffee break) and disclosed independently many times; this is just a matter of luck and effort for an alleged infringer to find a prior disclosure of a Columbus egg;
- or the description must contain enough details to allow reproducing the invention, otherwise the public has no evidence that the inventor owns the claimed invention.

Poor description not only breaks the patent contract. It infringes the patent law as well. We found in the [Guidelines to Inventorship Determination](#) of the McGill University this explanation: "Conception of an invention occurs when a definite and permanent idea of an operative invention, including every feature of the subject matter sought to be patented, is determined. Conception is complete when one of ordinary skill in the art could construct and work the invention as described and claimed without unduly extensive research or experimentation. The point in time when the invention, as defined by the claims, is conceived is when the inventor can describe the invention with its specificity, not when it is just a goal or research plan hoped for, or to be obtained hypothetically. In other words, an invention is complete and operative if the inventor is able to make a disclosure which would enable a person with ordinary skill in the art to construct or use the invention without extensive research or experimentation" [quoted in the document]. This is Canadian law but applies with some nuances elsewhere.

Examiners and courts hesitate to reject or invalidate software patents on the ground that their descriptions do not allow persons of the art to implement them:

- A software patent is a "cheap patent" whose patentee did not make extensive research or experimentation. Therefore even without any description a person of the art can implement the

Patent search

invention without extensive research or experimentation.

- In USA Bessen and Hunt found that "For software patents and business methods, the courts have largely eliminated this [enablement] requirement (Burk, 2002, Burk and Lemley, 2003 forthcoming)."

Enforcing a better enablement would not necessarily improve the quality of software patents.

- It is possible to describe in great detail a process even if it is not innovative. The only benefit is that "inventors" should wait up to the end of the project to patent whereas today "inventors" should patent as soon as they completed the specification to get the earliest filing date.
- A better enablement would increase the cost of writing a patent but this cost would remain marginal in the project cost. The software patent would remain a cheap patent.

Furthermore poor descriptions are usually not the consequence of a deliberate effort for seeking protection and at the same time keeping methods and processes secret. They are more often the consequence of lack of care.

Patent quality

When you browse patent databases you find evidences that the description section is not worse than the rest of a software patent. There are two types of claims: independent claims and dependent claims that refer to another claim. However we found claims referring to themselves (meaningless – they were not recursive!) We also found patents whose description did not allow a person of the art to make the same product but that described the patentee IT system (a gift for competitors). Patents are great examples of the interfacing problems described in Society and computing. A software patent requires the cooperation of three populations:

- Management who has the power but lacks skills and control means for the patenting process
- Programmers who have incentive to describe "inventions" but not for helping lawyers
- Lawyers whose objective is to keep under control the time spent on a patent

The wording and content of a software patent is a not really a concern for a company. A company is actually interested by another sort of quality that we could call unavoidable–ness. An unavoidable patent is a patent that other companies must license or that prevents competitors of offering the similar product. Unavoidable–ness depends on the width and location of the claim scope that cannot be easily measured. A patent that can be easily avoided is just a waste of time and money. Even a portfolio patent must prejudice competitor s interests to be useful in cross–licensing bargaining.

Reference metric

The only reliable way to measure the unavoidable–ness of a patent is to count the patents that refer to this patent. This is almost the same principle as the page popularity computed by Web search engines. This method is well described in Quantitative Methods of Research Evaluation Used by the U.S. Federal Government: "When a U.S. patent is granted it typically contains seven or eight references cited–U.S. patents on its front page. These references link the just issued patent to the earlier cited prior art, and limit the claims

Patent search

of the just issued patent. They point out where essential and related art already exists and therefore limit (are cited against) the claims of the citing patent. In one important respect patent citations differ from citations in a scientific paper. Front–page patent references are either put there by the examiner, or suggested by the applicant and his attorney and approved by the patent examiner, who is an expert in the art which he is examining. The net result of this is that a patent citation is undoubtedly stronger and more likely to be relevant to the subject area of the patent than a literature citation."

As we can see patent references are more reliable than paper citations and Web page links. However there are regional differences: "In the US, the applicant supplies a list of prior art references with their application, whereas in Europe the applicant is not required to do this and all prior art searching is done by patent office staff. Thus, front–page reference lists on U.S. patents benefit from contributions from applicants aware of the full technological context whereas in Europe, front–page reference lists are constructed by patent office staff naturally focused on the patent system."

"In the US, these procedural differences are reinforced by strong incentives. The U.S. rule of disclosure requires the applicant to disclose all known prior art references, and failure to do so is considered fraud on the patent office and can provide grounds to disallow the patent. Incentives to reference are strengthened in the U.S. in comparison to Europe by the heightened risk of patent litigation. The examiner is presumed to have examined a referenced document and to have decided that the invention was novel in relation to what was reported in the document. Therefore, a document referenced on a patent is much more difficult to use in court as evidence that the patent is invalid because the invention was not novel."

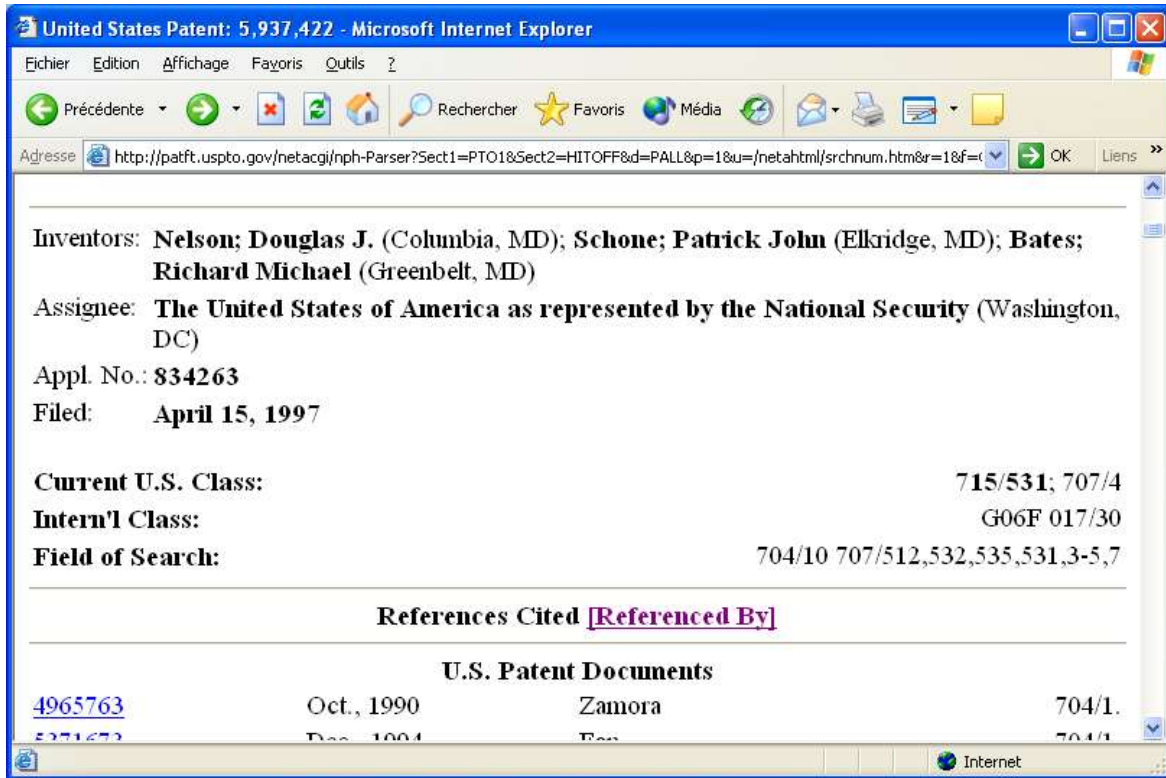
You can find this explanation on the USPTO site: "The U.S. Patent and Trademark Office cannot be aware of all PRIOR ART in a particular field of technology at the time a PATENT application is filed in that field. Therefore, the patent applicant is required to file an Information Disclosure Statement when the PATENT APPLICATION is filed, or shortly thereafter, that discloses the prior art that is most relevant to the CLAIMED INVENTION. The inventor is more likely to know the prior art in the field of invention than the PTO so the burden is on the inventor to educate the PTO about the prior art. Failure to comply with this requirement can invalidate any patent that issues on the grounds that the inventor committed a fraud on the PTO by not disclosing prior art of which the inventor was aware."

Unavoidable–ness impact

CHI Research wrote "Our investigation reveals a link between the technological strength of companies, measured quantitatively using patent citation analysis, and their subsequent stock market performance. The more high–impact patents that a company owns – i.e. patents that build on the latest technologies, are highly cited, and link extensively to scientific research – the greater its average share price increase." CHI Research even issued U.S. Patent No. 6,175,824 "Method and Apparatus for Choosing a Stock Portfolio, Based on Patent Indicators."

On USPTO to display the patents that refer to a patent click on the Referenced By link on the patent page:

Patent search



In the case of the NSA patent 5,937,422, which was one of the most read and commented patents you find 17 patents six years and a half after the date it was filed. 17 is a good figure. Most patents are not referenced at all.

We are not convinced that the CHI Research method is effective for choosing a stock in software industry because

- This method measures the performance of a company five years ago
- To convince the analysts a company must be able to show that it issued a sustained level of high-impact patents on much more than five years. Maybe only IBM is in this case.

Counting patent references is not a better metric for management:

- Maybe 95% of the software patents are not referenced. Few patents are referenced more than twenty times. From a statistical point of view how significant it is to have one reference instead of zero?
- A patent can be referenced by an applicant only when it is visible. In software industry we observe invention bursts when a new technology enables doing new things. These invention bursts are not longer than the secrecy period of patents (the time between the patent application and publication).
- How useful is it to know that the company did well or was granted no patent five years ago?

A company able to issue unavoidable patents certainly has a brilliant future. But this certainly requires a good deal of commitment, honesty and mutual trust. We explain in Society and Computing why such things can only happen by chance. If the programmer gets a fixed bonus per invention regardless of its virtues why

Patent search

should she confess that the thing she was working on did not give the expected result or is impractical? Why should she go beyond her assignment and overwork to really innovate when she knows that her efforts will not even be noticed? In the same way why the lawyer should try patenting with broad scope claims when she knows that broader claims increases the potential difficulties with examiners, the risk of opposition and claim invalidation for a patent that the customer will never read afterwards?

At this stage it is interesting to take a look at the University practice.

University patents

To look at the patent policies of Universities, search on Google with "university", "patent" and "policy" keywords. Though some Universities may be more generous than others these principles apply:

- All patentable inventions conceived or first reduced to practice by University members in the conduct of University Research shall belong to the University. On the other hand labor law says that inventions made by individuals on their own time and without the use of University resources shall belong to the individual inventor.
- In some Universities an expiration clause is defined. In cases in which the University has an ownership interest in an invention and either does not file a patent application within one year, or fails to make a positive determination regarding pursuit of a patent within a period of six months from the date of disclosure, all of the University's rights shall be reassigned to the inventor.
- A University does not necessarily patent even when the invention is found to be patentable and useful and to have a commercial interest. A University typically seeks a licensee first before filing a patent application.
- In the case of a patent owned by the University the inventor receives between 25% and 50% of the net royalties of the invention. Net royalties are gross royalties less expenses resulting from patenting and licensing the invention.

Universities define the following roles:

- Inventors
- Resources
- Licensing officer
- Marketing responsible for finding licensees

The inventor uses resources to reduce the invention to practice; the licensing officer addresses the legal aspects; marketing assesses the commercial potential and seeks licensees. The actual practice is somewhat different but at least there is an invention framework and inventors have an interest at inventing something with a commercial interest. Universities can patent inventions whose process or method is useful or unavoidable. The application of these principles to software patents in Industry is however not easy:

- A University looks for licensees. Its job is not to implement and sell. A company looks for a temporary monopoly. Its job is to implement and sell.
- A University looks for royalties. A company wants to exploit its invention. For a company, revenues from settlements, court decisions in infringement disputes and royalties are only a compensation for

Patent search

lost exploitation revenues.

- We showed in Society and computing that because software has a zero marginal cost a patent has a deterring effect. There are even anecdotal evidences that this effect is partly irrational. Today licensing a software process or method is usually not considered as an option. Therefore there are no customers for licenses.

Therefore even if the company put in place the same kind of compensation scheme as Universities the inventor will receive a fair share of the invention exploitation. However we believe that the interest of software companies is to put such schemes in place. The inventor commitment is the key factor for patent usefulness and unavoidable-ness.

Portfolios

"Software patents shifted from being relatively expensive during the early 80s to being relatively cheap during the late 90s." but the cause is a mentality change. Companies that felt exposed to lawsuits patented to reduce their exposure, which is consistent with another Bessen and Hunt observation " It appears that most software patents are acquired by firms in industries that accumulate large patent portfolios."

Accumulation of patent portfolios did not start with software patents. This may even be a legacy of Edison, who first designed a company for inventing. There are evidences that before WWI patents were already "understood to have important uses outside the protection of technology that the firm planned. In particular, they were an important strategic weapon in the long term struggle between a large firm and its equally large, equally well heeled rivals." [Found in One Hundred Years of solicitude: Intellectual Property law, 1900–2000 by Robert P. Merges; itself referring to Leonard Reich]. Quite interestingly one of the pioneers of patent portfolio accumulation was General Electric (a merger of Edison General Electric Company and Thomson–Houston Electric Company in 1892). You can find on the General Electric site: "By 1890, Edison had organized his various businesses into the Edison General Electric Company. The Thomson–Houston Company and the various companies that had merged to form it were led by Charles A. Coffin, a former shoe manufacturer from Lynn, Massachusetts. These mergers with competitors and the patent rights owned by each company put them into a dominant position in the electrical industry. As businesses expanded, it had become increasingly difficult for either company to produce complete electrical installations relying solely on their own technology. In 1892, these two major companies combined to form the General Electric Company."

At the beginning of the twentieth century the Society was more intellectual–property driven than now. Thomson–Houston and Edison were neither looking for economies of scale, nor buying brands or market shares. The merger was the solution to a deadly embrace problem. This was a serious issue: the dispute between Wright brothers and Curtis almost killed the American aircraft industry at the very time this industry had a tremendous development in Europe thank to WWI. The declining importance of Intellectual property after WWI may have been caused by the permanent war state up to 1989:

- Governments were omnipotent
- As far as defence was concerned the Intellectual property did not apply
- Innovation was primarily the consequence of military research that gave us aircrafts, electronics, computers and even Internet
- Cooperation was a kind of patriotic duty

Patent search

Since 1989 the World is again in peace but things changed since 1914. When Edison used patents to conquer the electrical industry today large firms issue patents to secure their market position. Software patenting is more defensive than offensive. Bessen and Hunt note: "Overall, software patents are more likely to be obtained by larger firms, established firms, U.S. firms, and firms in manufacturing (and IBM); they are less likely to be obtained by individuals, small firms, newly public firms, foreign firms, and software publishers." Bessen and Hunt found that: "The manufacturing sector acquires 69% of software patents, but employs only 10% of programmers and analysts; software publishing and services (including IBM) acquires only 16% of software patents, but employs 42% of programmers and analysts." They also found that software publishing acquire 6% of software patents, other software service firms, excluding IBM acquire 2% of software patents.

IBM is the giant of Intellectual Property and not only for software patents. IBM received 3,288 U.S. patents from the USPTO in 2002 and 3,415 in 2003. Over 1,250 of these patents were software patents of a quality above the average.

Patent accumulation used to be a weapon in the long-term struggle between a large firm and its equally large, equally well-heeled rivals. In the software industry there is no economy of scale and the rival of a corporation can be a short-lived small company. Furthermore the Economy is global. Software patents cannot be credible as an offensive weapon in a context where:

- identifying infringing companies is costly,
- suing a smaller company, especially in another country may damage the firm brand,
- other large firms also have a large number of patents, which creates a situation of mutual infringement.

Patent field

The software patent space is filled in two ways:

- Horizontally by IBM and other software publishers for core technology patents
- Vertically by market leaders for market-specific applications of computing. Profitable markets dominated by large companies have a high patent density. Less profitable markets in which companies are smaller may be almost patent-free.

On markets with a high patent density patent wars and Intellectual Property-driven mergers could happen in the same way as at the beginning of twentieth century. This would maybe have already happened if market leaders had been able to file good quality patents and to enforce them effectively.

For core technology patents wars and mergers are unlikely to occur.

First core Software patents are profitable for patentees only in two scenarios:

- Strong patent owner against strong patent owner. In this case patents secure market shares and limit competition between large firms that sign transactions and cross license their patents.
- Weak patent owner against strong patent owner. In this case a patent can be an extremely profitable investment because the strong patent owner must either pay license fees as illustrated by 5,838,906

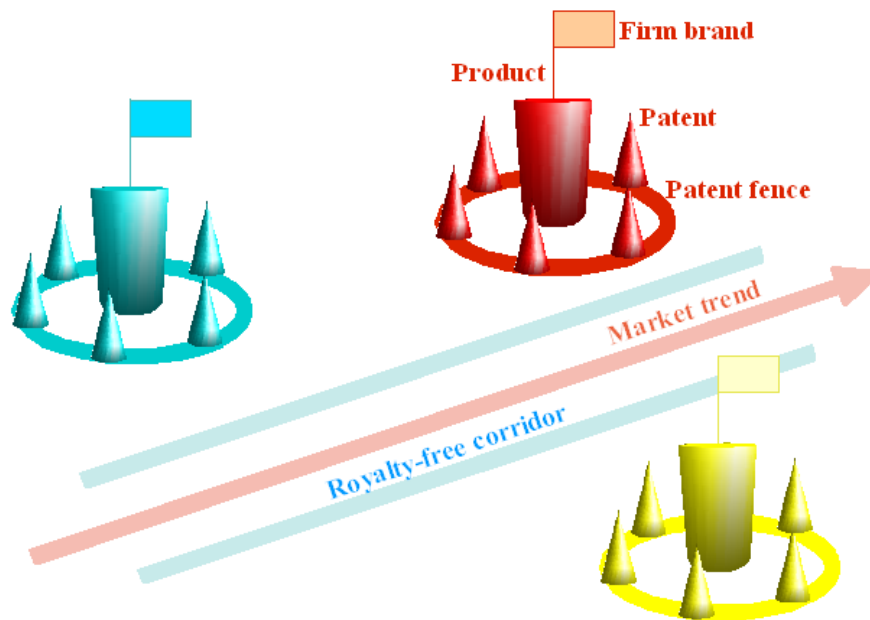
Patent search

(Eolas) or buy the weaker company.

Today the other scenarios are not profitable. A weak patent owner cannot make money at identifying small infringers and seeking settlements or suing them as we have seen with PanIP. A strong patent owner cannot easily ask license fees to small companies without hitting the headlines and facing serious problems. The Unisys story with the LZW compression method (patent 4,558,302 on USPTO; if you do not remember read this [reminder](#)) illustrates this issue.

Second customers are primarily concerned by switching costs:

- They prefer solutions based on royalty-free based standards
- They use patented solutions only if the solution is implemented in a perennial product with a good value for the price.



The market tends to follow standards. The reference implementation of the standard and the Open Source implementation sit in a royalty-free corridor around the standard. Commercial products implement the standards and additional functions. These functions are protected by patent fences that prevent competitors of implementing a super set of (1) the standards (2) the functions of the competitor product (3) the functions of the firm product. Large publishers make research and sponsor standards. They can patent in areas where a market could emerge or move like a property developer can buy estate where a new airport, highway or fast railway will be built. Patent fences can overlap. In such case firms seek a settlement or cross license their patents. Therefore the core technology patent space is a desert with high patent density areas in the vicinity of royalty-free corridors.

Intellectual Property strategies

Software intellectual property is an issue for everybody involved in using or producing software. However, so far, only large firms with legal departments paid the needed attention to this issue. Smaller companies tend to see software patents as an unnecessary burden; refuse to spend time on Intellectual Property issues and when they realize that there is a danger at doing so lobby to ban software patents. We believe that it is a short-minded view for the following reasons:

- You can doubt the value of patents for the progress of useful arts or as an incentive for innovation but if you accept the patent system you cannot easily exclude software inventions from that system. Before software patents were officially allowed (1997 for USA) companies patented apparatus made of a computer and of a process or method. The software patent law probably has to evolve but it is here to last.
- Authors rightfully noted that the lack of software patents did not prevent the tremendous development of the software industry. However this development happened in a scientist community. Since that time the computer society has changed. Now it may need Intellectual Property to innovate. Most companies analyze competitor products and copy their most successful functions. The safest way to succeed is to implement, promote and support better what others have already done. This is also the easiest way. Without experimentation phase a development is easier and can be released on time. This is actually easy enough to allow countries with lower labor costs to deliver the same clones with the same high quality and tight schedules. The current lack of protection may actually kill innovation in a domain where there are still many things to invent.

In this section we focus on smaller companies. Large firms develop sensible patenting strategies with the help of lawyers and consultants. They need more data than we can possibly present here. However this section may also be useful for people who work in these large firms on a business uncovered by the patent strategy and understand that ignoring Intellectual Property for any activity of the large firm is a suicidal negligence.

A company has three ways to protect its intellectual property:

- Secrecy
- Patenting
- Publication
- Open Source

Secrecy

Secrecy is the default Intellectual Property policy. If nothing is done programs, documentation, ideas remain hidden in the hard disks of programmers and sometimes in the company repositories. Default secrecy is free but has several drawbacks:

1. The management does not know what are the software assets of the company
2. Code and design reuse is impossible except for the original author
3. In case of patent dispute the company may be unable to demonstrate that it knew or used the disputed invention before the patent was filed

Patent search

To address the first issue the management may ask for additional reporting and for a software inventory. To address the second issue the company may put in place a document management system (DMS), forums and mailing lists with search means and build an Internet-like internal system. This system may be expensive to set up and operate and is usually ineffective for the following reasons:

- In a company where secrecy is enforced the best strategy for a programmer is to keep secret her knowledge and findings. She may be laid off. In such circumstances less she disclosed in her current job more she can reuse in her new job. She also has fewer chances to be fired if nobody can easily replace her. [I think however that her long term interest is to disclose her findings to her peers and managers because she needs feedback to make progress.]
- These DMS, forums and mailing lists are often underused. To get users these tools (1) must contain enough entries (2) answer user questions (3) not have competing means: if the person who can answer is at the next office why recording the question? Such tools are only effective when they are the most convenient medium for asking a question or sending a message because of language barriers and power distances.

To address the third issue companies can deposit their program sources and documentation in escrow.

Escrow

Escrow serves three functions:

1. A customer may ask that the provider deposits the components agreed upon in the contract in escrow. Then if the provider goes bankrupt or abandons the product the customer can get the code and documentation from the software escrow. The customer may designate its escrow or agree to use the provider escrow.
2. A company can make deposits in an escrow to prove the dates when it made the inventions used in its products.
3. A company can make deposits in an escrow to prove that former employees stole a part of its intellectual property

With escrow a trusted third party can assert that the company owned sources, documentation and implied processes and methods at a given date. Because the company did not publish its processes or methods someone else can patent it. The patentee cannot ask for patent fees if the company can prove that it owned the invention before the filing date. However because it has no knowledge of the owned process or method, the patentee cannot find that the company had possession of the invention and invisibly abandon legal actions against the company. Therefore the company must address the patentee action, usually hire a lawyer and often go to court. Remember that the plaintiff complains to the richest party. If the company is a small software publisher that sells to large firms the plaintiff may choose to sue the largest customers. Therefore the company may suffer from the dispute even it is able to present convincing evidences of its ownership. Some Internet documents suggest that prior possession is not necessarily considered in all countries and that a company must either patent or make public in some places.

Escrow service is relatively inexpensive. You can find many escrows on Google with the keywords "software", "deposit" and "escrow".

Sources

A common attitude is program fetishism. Program fetishists put only sources in escrow and consider that the major risk that they face is disgruntled employees that steal programs.

- Their attitude is inconsistent: Why spending so much money in marketing and analysis if only programs matter? In reality a good product combines a good definition, a clever design and a decent coding and the company asset is made of the product, of its documentation and of a deep knowledge of the product. In that list code sources are the most tangible and the less important item.
- By default programs belong to the company and not to the programmer. Companies must ensure that employees do not reuse elsewhere programs written by other programmers. Trying to ensure that programmers do not reuse elsewhere programs they have written may have an adverse impact: companies depend on the programmer cooperation to keep the design documents and the documentation synchronized by the programs. They also depend on the programmer memory to retrieve relevant prior art. If she feels that she has not the right to reuse her code a programmer may not disclose her most important findings.
- Program fetishists may be right when the communication between management and programmers is so bad that the product definition and design is not accurately tracked. This situation may harm the company in case of dispute. A design document that describes a key patented claim can be enough to put a defendant out of danger. Proving that a piece of code actually implemented the process or method implies involving third party experts and requires more time and money.

A company must put in escrow:

- Source code
- Manuals (especially the user and reference manuals)
- Trainings
- Documentation and screenshots of the user interface, especially at an early stage when manuals and courses are not yet written
- Product definition and design documents. This is important to also put in escrow abandoned design documents.
- Analysis documents

Conclusion

Secrecy is what you get when you do nothing and is cheap. Secrecy is convenient to hide little secrets like "borrowed" and plagiarized code and poorly written and tested programs. More important secrecy does not help a company to assess its strengths and weaknesses and maximizes its exposure to patent infringement liability.

Patenting

Patent search

Issues

To patent you need competent resources or help. Patenting is not an option if

1. nobody in your company has a working knowledge of patent writing and prosecution, or
2. you do not have a legal department and a full time contract with a patent attorney, or
3. you have an incubator that does not provide such service, or
4. you do not partner with an experienced university.

If your company has a patent portfolio strategy you can patent almost everything novel and inventive and you should turn your attention at filing good quality patents in cooperation with your lawyers.

Otherwise consider that patenting like programming is a job with a sharp experience curve. Your first patent will cost you more than the second and so forth. Therefore filing a first patent is a strategic decision. For this decision to be profitable you need to file other patents and to dedicate a significant part of your resources to patenting. When we analyzed 5,838,906 we found that the assignee, a small company called Eolas owned at least seven patents and that 5,838,906 was indeed the second Eolas patent.

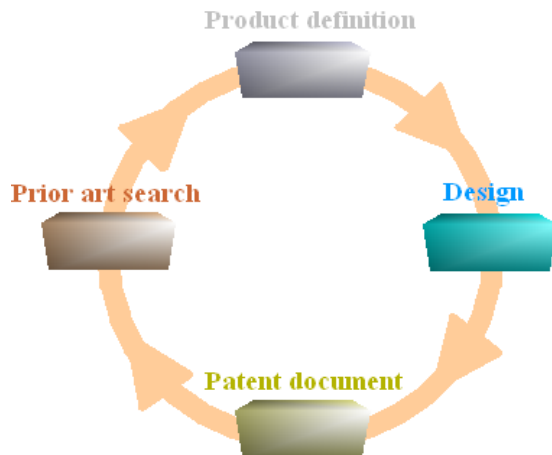
You must be honest about your position on the market. If you are a me-too company you probably should not patent. You need to be a technological leader that (1) by chance had to implement something substantially different of what already exists (2) has good chances to have again to implement novel processes or methods because you will need to file subsequent patents. Even if your invention is really novel, non obvious and useful the software industry can follow a different path and ignore your patent. Patenting is a risky business.

Eventually you have to go fast. We have seen that browser controls and plug-ins would have invented anyway between 1994 and 1995. This is not unique to software inventions. Without Bell and Edison the telephone and the electric bulb would have been invented anyway when they were invented. At a given time the technology is mature and different teams work independently on the same idea. You must quickly and thoroughly assess the novelty and the potential of the invention, write a patent draft, explain the invention to lawyers, check their rewording. It does not finish when the application is filed. The examiner can require changes and even reject the application. This is not easy in a small company. Filing a patent is urgent but to the opposite of product deadlines there is nobody to remind you that it is urgent.

Introduction

Here is an introduction to patent writing. This explanation is useful even if you do not plan to file a patent. Before starting a new project you should look for possible infringement. For this search as we have seen in the toolkit section the best is to create a reference text that looks like a patent and to use this text to find patents with the same word frequencies. This work complements a design. It points out the differences between your design and existing products and lists the problems that you were trying to solve. You should refine your product definition according to your finding in prior art:

Patent search



We may have to iterate the product definition, the design, the patent writing and the prior art search. The design answers the feasibility question and the patent search the legal question. For prior art search you do not need to write a full patent. You may focus on background, prior art and summary.

According to the USPTO guidelines a patent is made of three sections:

- An abstract
- Claims defining "in words the boundaries of the invention so that the public will know what the invention is and can avoid infringing it." Claim wording is a patent attorney job.
- A description

The reference guide for writing a patent is the Manual of Patent Examining Procedure (MPEP). Read in particular the Chapter 2100 Patentability.

Abstract

We found this useful definition:

"A patent abstract is a concise statement of the technical disclosure of the patent and should include that which is new in the art to which the invention pertains. If the patent is of basic nature, the entire technical disclosure may be new in the art and the abstract should be directed to the entire disclosure. If the patent is in the nature of an improvement in an old apparatus, process, product or composition, the abstract should include the technical disclosure of the improvement. [...] If the new disclosure involves modifications or alternatives, the abstract should mention by way of example the preferred modification or alternative.

The abstract should not refer to purported merits or speculative applications of the invention and should not compare the invention with the prior art.

Where applicable, the abstract should include the following:

1. if a machine or apparatus, its organization and operation; [...]

Patent search

2. if an article, its method of making;
3. if a process, the steps."

Claims

The function of claims is explained in 35 USC 112: "The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention."

Here is an example of claim:

"A computer-implemented method of searching for an item in a plurality of independently operated electronic auctions interconnected by a computer network, each electronic auction having an associated data repository, the method comprising:

- receiving input identifying an item; and
- instructing a software search agent to search for the item on the computer network in the respective data repositories of one or more of the electronic auctions."

It has a preamble and a number of elements (words or phrases) that describe the claim limitations. There is infringement when another party use a system with each and every limitation of a claim. So the inventor must include the absolute minimal set of elements that differentiate a new invention over what came before. If US patents contain many claims this is to maximize the chances to get a match. When a claim contains many limitations this is usually because the examiner rejected the first version of the claim for lack of novelty (it was anticipated by prior art) or obviousness and because the applicant had to add additional limitations to get a patent.

This claim has a function, which is to search "an item in a plurality of independently operated electronic auctions interconnected by a computer network" and two steps:

1. "receiving input identifying an item; and
2. instructing a software search agent to search for the item on the computer network in the respective data repositories of one or more of the electronic auctions."

Such claims are called means or steps plus function claims. They are so important that 35 USC 112 describes them: "An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof."

So the precise scope of such claims is determined by the claim and by the description. There was a trial for infringement of the patent that contains the claim above. The court determined the meaning of words like item and also whether the steps recited a sequence. To learn more about this case you may look at the [Business method page](#). For details about the construction of means plus function claims you may look at <http://www.uspto.gov/go/og/con/files/cons089.htm> and <http://jip.kentlaw.edu/nart/2004/A-2.htm>.

Patent search

As implicitly stated by 35 USC 112 a claim may also recite structure, material, or acts. In this case the claim loses its function limitation. A claim for a popcorn dispenser was rejected because the claimed structure was the same as the structure of an oil can.

Description

The description is made of:

- Title: short, brief and specific
- The field of the invention: a broad statement that should start with something like "The present invention relates to". In case of 5,838,906 it was "This invention relates generally to manipulating data in a computer network, and specifically to retrieving, presenting and manipulating embedded program objects in distributed hypermedia systems."
- Background information: what people will need to understand, search or examine your invention. Do not assume that the reader has a prior knowledge of your job. Present your activity like you would do with your neighbor or your mom.
- Prior art: Problems that inventors have faced in this area and how they have attempted to solve them. You must search patents and prior art in the area because (1) your invention may have been already invented (2) as we have seen in the reference metric discussion you must cite the prior art and patents you are aware about. This is important because a patent shall be presumed valid as said in 35 USC 282 but "while the presentation at trial of a reference that was not before the examiner does not change the presumption of validity, the alleged infringer's burden may be more easily carried because of this additional reference." So an alleged infringer can more easily demonstrate that your patent is invalid if you omit relevant prior art in your reference list. In the prior art section briefly describe how these prior art and patents are working and the problems that they do not solve.
- Summary of the invention: How your invention solves one or several of these problems. What you are trying to show is how your invention is new and different.
- Drawing and screenshots
- Detailed description: Must describe the invention parts in sufficient detail so that someone could reproduce at least one version of your invention. Describe each part of the invention, how parts fit together and how they work together. For a process describe each step, the initial state and the result.
- Example of an intended use for your invention (best mode)

When you write your first patent document the most important problems you face is that you do not have the right mindset and you were not prepared to describe your work in such terms. A patent is neither a marketing nor a analysis document. Use the same sentence structure and the same verbs as in IBM patents (on the USPTO advanced search an/IBM). Check that you do not use words like innovative, powerful, successful, synergy, strategy and project. Do not use the word product outside the background and the prior art section.

Publication

If patenting is not an option you can make your invention public. Your competitors will be able to use it but not to patent it because a public process or method is prior art. This can actually help in your business. Software patents are repellent to competitors and if you are the only company doing things in one way your

Patent search

product will not sell. If you made your invention public competitors will use it and because you have some advance you should increase your market shares.

Mailing list

As we have seen in the 5,838,906 case a public mailing list whose readers are concerned by the invention matter is sufficient because "An electronic publication, including an on-line database or Internet publication, is considered to be a 'printed publication' within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates." A mail is probably the cheapest and fastest way to make public an invention.

On the other hand the court of *MercExchange v. eBay* found that a newsgroup posting does not have the status of a printed publication. It gave two reasons:

1. There was no guarantee that any party would have been granted permission to join in the newsgroup. In our view this reason is not valid for newsgroups but it may be for some forums.
2. Newsgroups have a retention period after which they are removed. This period depends on the disk space allocated to the newsgroup and is therefore variable. So it is difficult to find a posteriori how long a message remained posted.

So to make public an invention we recommend to not use newsgroups and forums whose read access is password protected.

According to most national laws a mailing list can be in a country and a language different of the court country and language. However, to be on the safe side, make your disclosures in English.

Article and presentation

There are two other solutions that you can combine with mails to mailing lists:

1. Publish an article on a newspaper or on an electronic publication. Writing an article is not obvious but is simpler and faster than writing a patent. An article has to be a story of around 3000 words that must capture the reader attention at the beginning and keep it up to the end. You can use a plan background – prior art – description. There are side benefits at writing articles. A customer who has liked your paper is more likely to buy your product. In case of newspaper publication you can use reprints to target technicians who are generally insensitive to traditional marketing material. In case of electronic publication you can link your site to the article.
2. Present the invention to a conference. It may imply joining an organization and paying for membership. You also have travel expenses. The presentation cost is not necessarily lower than the patenting cost. If you already participate to an organization the presentation can be a good solution because it allows more flexibility than an article. You can also use the background – prior art description plan.

Both solutions produce prior art of premium quality but they have a common drawback, which is to take time.

Patent search

You usually submit a proposal either to the publisher or to the organization. Your proposal may be approved with some amendments one month later. Then you send your article or presentation. Next the publisher or organization has to find a slot for you in its next issues or meetings. In case of newspaper article or presentation it takes between four and six months. Electronic publication is faster. According to most national laws the article can be written or the presentation given in a country and language different of the court country and language. However, to be on the safe side, make your disclosures in English.

Web site

A publication on your Web site is usually not considered as an Internet publication because you cannot prove the last modification date of your publication. There are however a couple of cases worth to consider:

- If your Web site is well indexed by search engines and is making 100,000 visits per month when your competitor files its patent, there are good chances that (1) at least in USA, you competitor will have to cite your site and show how its invention is different of your process (2) the examiner will check your site and reject the application. In both case you avoid a lawsuit.
- There are Internet archives. Maybe the most famous is <http://www.archive.org/>. If your site is popular your pages may be recorded every month on this site and this facility is routinely used by courts and examiners. Take care of not changing the page locations because this can confuse the Internet archivists.

A popular and content-rich web site is not only a contribution to Internet knowledge and a way to attract programmers. It can also limit your exposure to patent lawsuits.

Disclosure service

Some companies offer a technical disclosure service, typically on Internet. An example is <http://www.ip.com>. For a single publication the price is \$155 (November 2003). This kind of service is interesting because you do not have the travel and membership fee associated to a presentation and you do not need the writing skills needed for an article. You can just publish a cleaned up version of a design document.

Open Source

The most radical solution is to release your core code with an Open Source or free (GPL) license.

The best location for this is probably [SourceForge](#) because this site is well known of persons concerned with most computer arts. Two key points are:

- "Only Open Source projects may be hosted on SourceForge.net. Projects are expected to meet the requirements of the [Open Source Definition](#) and abide by the spirit of that definition. With this in mind, it is important to understand that projects hosted on SourceForge.net are expected to release their source code (as per Clause 2 of the Open Source Definition) — projects may not make binary-only releases."

Patent search

- "When registering your project, you will be expected to provide a full description of your project, information about which Open Source license under which your project will be released, and the name of your project."

SourceForge registration and hosting are free.

Background

According to the OSI definition an Open Source license must conform to the following rules:

1. Free Redistribution. The license shall not restrict any party from selling or giving away the software.
2. Source Code. The license must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost preferably, downloading via the Internet without charge.
3. Derived Works. The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor. This clause exists to prohibit license traps that prevent open source from being used commercially
7. Distribution of License. This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.
8. License Must Not Be Specific to a Product
9. The License Must Not Restrict Other Software. GPL is conformant with this requirement. Software linked with GPL libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.
10. The License must be technology-neutral. This clause actually means that the product can be distributed in FTP or on CDROM.

The OSI definition comes actually from Debian. Debian is a Linux distribution and the authors of the license definition had in mind to give a definition enabling a business including:

1. Stable rules
2. Third-party distribution
3. Newspaper distribution
4. A compelling offer to users. People want free products and at the same time to release commercial products.

This section aims to help you choosing a license. The two most important families of Open Source licenses are BSD-based licenses and GNU licenses.

The Berkeley University had developed a number of programs under this license that were used in institutions and industry. This license addresses three needs:

Patent search

1. Disclaimer: Berkeley doesn't want to be liable to prosecution for bugs in its programs
2. Copyright: Berkeley wants to enforce its copyright to show that it wisely use public funds
3. The name of the university and of the author may not be used to endorse or promote products derived from this software

These needs were typical of 70 s and 80 s University needs. The following products have a BSD-based license:

- Java
- Apache products
- XFree 86 and other licenses used with XFree 86

Quite interestingly the BSD license is titled "Berkeley-based copyrights". The most important constraint of this license is that you have to advertise the contribution of the publisher. However there are even more user-friendly licenses like the ACE / TAO and the MIT license used with some changes by Expat. Use a BSD-based license when you want to facilitate the life of your users, for instance:

- You offer a Web service. You want to provide free tools to your users for using your service.
- A competitor took all the market but this market is so important for you or so profitable for your competitor that you are ready to give away your code to come back on this market

As far as Intellectual Property is concerned you do not need to use a BSD-based license and GNU licenses usually serve better your needs because they prevent users to derive non-free products from your code. We use free here because the father of GNU, Richard Stallman rejects the "Open Source" term and prefers the "free" word: "When we speak of free software, we are referring to freedom of use, not price." Note that in the rest of this document when we talk about Open Source we mean Open Source or free. The two most important GNU licenses are the GNU General Public license, GPL in short and the Lesser General Public License, LGPL in short. These licenses enforce the copyleft principle, which states that you must release a modified [L]GPL program with the same license as the original program. Things are however more complex because of derivative works.

First the difference between GPL and LGPL is that the LGPL license allows linking non-free programs to LGPL libraries whereas GPL doesn't allow it. Here is the rationale: "When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library." More precisely the LGPL permits to programs containing no derivative of any portion of the Library, but designed to work with the Library by being compiled or dynamically linked with it, to be a non-free product. An executable statically linked with the library contains the library and therefore should be released under a [L]GPL license but objects used to build the executable that don't contain portions of the library don't need to.

GNU license put no restriction on the program use. The GPL license says: "Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program)." The restriction on the

Patent search

output is clarified by the GNU FAQ: "If the user uses your program to enter or convert his own data, the copyright on the output belongs to him, not you. More generally, when a program translates its input into some other form, the copyright status of the output inherits that of the input it was generated from. So the only way you have a say in the use of the output is if substantial parts of the output are copied (more or less) from text in your program. For instance, part of the output of Bison (see above) would be covered by the GNU GPL, if we had not made an exception in this specific case."

GPL is the license of GCC and most Linux programs. LGPL is the license of the GCC library.

The GNU wrote a couple of documents of interest for choosing a license:

- An list of Open Source and free licenses called [Various Licenses and Comments about Them](#)
- Explanations about free software and OSI that you find [here](#)
- A [comment](#) about the BSD license

Rationale

In some circumstances releasing your core code with an Open Source or free (GPL) license is a rational choice from an economical and legal point of view. When a new market emerges a patent and publication race starts. If you never patented and never published you have almost no chance to win. If you distribute your core product and documentation with an Open Source license starting with the alpha version you have a good chance to win the race even if you have less resources than your competitors. In that respect Open Source acts as publication. Your competitors will be able to use your invention but not to patent it because your implementation is prior art. Obviously your competitors can also use your implementation. This is not necessarily a bad thing:

- You should be able to convince your customers that you know better your code and that you can support it
- If your competitors use your product you control a part of the market

Open Source has other strong points:

- When you release your core product with an Open Source you make a superior proposal to your customer. As we have seen above customers often require a source deposit in escrow. The escrow checks that the deposit conforms to the contract but the customer has no access to the code.
- Customer s programmers may test your core product and use it to make some prototypes and development before signing a contract with you.

Sales

You still can sell a product that uses the Open Source part and contains other facilities like configuration and administration user interfaces. The differences between the core part and the whole product are the prerequisites. The core part requires an intimate knowledge of the product technology whereas the whole product can be administrated and used by almost anybody. While they do not usually contain a significant

Patent search

amount of Intellectual property the extra facilities are essential to the product success and frequently more expensive to implement than the core product.

There are other solutions:

1. You can use a license that does not grant the right to use the product for free when you get paid for using it. An example is the Caucho Developer Source License used for the Resin Application server. Note however that such licence may not conform to the OSI definition.
2. You can use a dual license like MySQL. On the download page of MySQL users are given the choice between a GPL version provided "as is" without any warranty and a commercial version the user needs to make commercial derivatives of MySQL or to get a warranty or support. The CEO of Trolltech, an early adopter of dual licenses, explained: "Berkeley DB was originally available from UC Berkeley under the popular open source Berkeley Software Distribution (BSD) license. When we decided to form a business around Berkeley DB, we looked at the BSD license closely. The BSD license would give our customers flexibility. It would allow them to freely embed Berkeley DB in hardware or software products and distribute their product without restriction. However, the BSD license would also allow our customers or any developer to take our code and create their own products without our permission or payment. We would create our own competition. We decided to begin the business with a dual-license strategy [GPL and commercial]." Dual license works well in the common case in which the GPLed product has a very small developer community outside the company that support it.

You may read that Open Source and free software are untested in court and therefore unproven. As we have seen in the Patent issues section software Intellectual property will continue to evolve, especially regarding how much care must be taken by a company to get an enforceable patent. Furthermore this is only half the truth. From time to time free software goes to court. Before considering Open Source legal issues we must first briefly explain what Copyright and trademark are.

Copyright

A copyright comes into effect immediately, as soon as something that can be protected is created and "fixed" in some way. A copyright is an unregistered right. In most country the copyright notice is not even necessary. Writing the copyright symbol (©) followed by the company or organization name, the date and "all rights reserved" warns others against copying the copyrighted stuff, which may strengthen your position in case of dispute.

There is no incompatibility between Open Source licenses and Copyright. You actually must claim to be the owner of your code to license it with an Open Source license.

You can find the Copyright law of USA at <http://www.copyright.gov/title17/>. Copyright law is covered by Trade-Related Aspects of Intellectual Property Rights (TRIPs) like patent law. Governments try to harmonize their Copyright law like their patent law.

Trademark

A trademark is any sign that can distinguish the goods and services of one trader from those of another. For more information about trademark you can read our [trademark page](#).

MySQL v. Progress

The case opposed MySQLAB (plaintiff) to the NuSphere subsidiary of Progress (defendant). The two companies had formed a "strategic" alliance. Here is an excerpt from the press release: "To create an open source business, Progress Software has formed NuSphere, a Progress Software Company, to distribute and support the NuSphere MySQL open source database. Progress Software will be providing up to \$2.5 million (US) to help further the development of MySQL. NuSphere will extend the core development team's efforts by creating NuSphere MySQL; a multi-platform integrated distribution of MySQL and related open source products. NuSphere will provide commercial support, consulting, and training for customers deploying MySQL as part of their web infrastructure. As a single, global source of integrated services for MySQL, NuSphere will provide support from three main support centers in the U.S., the Netherlands and Australia. NuSphere chose to support MySQL because of its position as the market-leading open source database for web infrastructure. MySQL is especially well suited for capturing web site traffic, providing authentication and authorization services, and content personalization. The MySQL database is the first widely used database product released under the Free Software Foundation's General Public License (GPL)."

We understand that at some stage NuSphere released a database product called Gemini statically linked to MySQL's code without the accompanying source code. NuSphere did another thing that may have triggered the dispute. Though MySQL has a GPL license the original MySQL Web site was <http://www.mysql.com>. NuSphere opened a <http://www.mysql.org> Web site.

MySQL asked to a Boston court to issue a preliminary injunction to bar Progress, and its subsidiary NuSphere Corp., from using MySQL's trademarks in their products and from distributing any software that included MySQL's self-titled database program. MySQL alleged that Progress failed to include the source code for a software tool NuSphere built and included in its MySQL Advantage package, which included MySQL's database. GPL requires that any software built using components released under the GPL must also include the source code of the new application. Eben Moglen, General Counsel to the Free Software Foundation (FSF) testified as an expert witness. Here is his affidavit. Ed Kelly and John Palfrey reported the meeting to Tim O Reilly: "Judge Saris was surprised at how many people were there to hear the arguments on a motion for a preliminary injunction barring use of the MySQL mark by NuSphere. She noted the almost religious zeal of the open-source community. She also noted that both parties came loaded for bear and that MYSQL had flown in David Axmark from Sweden [MySQL is a Swedish company] and both parties had expert witnesses prepared to testify."

Judge Sarris refused to allow arguments to expand beyond the trademark dispute. Though she did not issue any orders in court, Saris made clear that she intended to issue a preliminary injunction against NuSphere's use of the marks in question and that she was not going to issue a preliminary injunction in the more complicated matter of the use of the MySQL code and terms the GPL. She urged both parties to come to an out-of-court settlement. From our perspective the most important is her comments about the GPL license:

Patent search

- GPL was treated as any other license would be in a software context
- Judge Saris seemed to focus on the question of whether Gemini constitutes an independent or a derivative work (does "linking" to another program equate to creation of a derivative work?) and whether the harm caused to MySQL met the irreparable standard.

ComputerWire reported the story with different words and a different title: Judge blocks route to GPL legal test case.

Then Judge Saris issued a preliminary injunction. In the ruling she barred Progress and NuSphere from advertising or selling any products using the MySQL name and from registering or operating any Web sites that include the MySQL name. The ruling did allow Progress to state that its products interoperate with MySQL. Saris, however, declined to rule on the GPL issue, stating that MySQL had not shown a likelihood of success in the matter or irreparable harm, two prerequisites for the issuance of a preliminary injunction. MySQL seems to have the better argument in the matter, Saris wrote, but said that the release of source code by Progress in July 2001 may have solved the problem. Because of MySQL's central place in NuSphere's product line, barring the company from using the software could substantially hurt NuSphere, she said.

Eventually MySQL and NuSphere concluded a settlement on November 7, 2002. We read the settlement that way: "The settlement resolves all outstanding issues between the two companies including [...] assignment to MySQL AB of copyrights for all NuSphere contributions to the MySQL program", which means that (1) the GPL license was extended to NuSphere contributions (2) the copyright of NuSphere contributions was transferred to MySQL AB. Again ComputerWire reported the settlement is a different way in MySQL, NuSphere Settle GPL Contract Dispute.

This case shows that:

- The irreparable harm prerequisite may bar preliminary injunctions in further cases. If you give an application for free you cannot claim that you suffer irreparable harm when a user infringes your license whereas the user can prove that the injunction hurts its business.
- The GPL may be found partially illegal. Much of Saris questioning surrounded whether the application could operate without MySQL. A company could for instance develop an application that can use MySQL, PostgreSQL and Oracle for database access. Then according to the copyleft principle this company should release its code under GPL but the judge would probably find that the company does not have to. Richard Stallman wrote: "Using the ordinary GPL is not advantageous for every library. There are reasons that can make it better to use the Library GPL in certain cases. The most common case is when a free library's features are readily available for proprietary software through other alternative libraries. In that case, the library cannot give free software any particular advantage, so it is better to use the Library GPL for that library. This is why we used the Library GPL for the GNU C library. [...] However, when a library provides a significant unique capability, like GNU Readline, that's a horse of a different color. The Readline library implements input editing and history for interactive programs, and that's a facility not generally available elsewhere. Releasing it under the GPL and limiting its use to free programs gives our community a real boost." From our point of view this is not only a matter of interest. If users have an alternative to your product for addressing a certain need then you should use LGPL.

SCO v. IBM

SCO filed a suit against IBM in the district court of Salt Lake City through its Caldera subsidiary. SCO wrote: "IBM has breached its own obligations to SCO [the Unix owner], induced and encouraged others to breach their obligations to SCO, interfered with SCO's business, and engaged in unfair competition with SCO, including by misusing and misappropriating SCO's proprietary software, incorporating (and inducing, encouraging, and enabling others to incorporate) SCO's proprietary software into open source software offerings and incorporating (and inducing, encouraging, and enabling others to incorporate) SCO's proprietary software into open source software offerings.

The proprietary code SCO is talking about is primarily UNIX and the Open Source software SCO is talking about is primarily Linux. SCO, summarizes the UNIX story: "AT&T Technologies originally licensed the UNIX operating system software code to approximately 30,000 software licensees, including defendant IBM, for the UNIX operating system software source code, object code and related schematics, documentation and derivative works (collectively, the UNIX Software Code). To protect the confidential and proprietary source code information, these license agreements, as detailed below, contained strict limitations on use and dissemination of UNIX Software Code. When SCO acquired the UNIX assets from Novell in 1995, it acquired rights in and to all (1) underlying, original UNIX software code developed by AT&T Bell Laboratories, including all claims against any parties relating to any right, property or asset used in the business of developing UNIX and UnixWare; (2) the sale of binary and source code licenses to various versions of UNIX and UnixWare; (3) the support of such products and (4) the sale of other products that are directly related to UNIX and UnixWare."

SCO explained the IBM strategy in this way: "In the process of moving from product offerings to services offerings, IBM dramatically increased its staff of systems integrators to 120,000 strong under the marketing brand IBM Global Services. [...] IBM sought to move the corporate enterprise computing market to a services model based on free software on Intel processors. By undermining and destroying the entire marketplace value of UNIX in the enterprise market, IBM would gain even greater advantage over all its competitors whose revenue model was based on licensing of software rather than sale of services. To accomplish the end of transforming the enterprise software market to a services-driven market, IBM set about to deliberately and improperly destroy the economic value of UNIX and particularly the economic value of UNIX on Intel-based processors."

About Linux SCO wrote: "Prior to IBM involvement, Linux was the software equivalent of a bicycle. UNIX was the software equivalent of a luxury car. To make Linux of necessary quality for use by enterprise customers, it must be re-designed so that Linux also becomes the software equivalent of a luxury car. This re-design is not technologically feasible or even possible at the enterprise level without (1) a high degree of design coordination, (2) access to expensive and sophisticated design and testing equipment; (3) access to UNIX code, methods and concepts; (4) UNIX architectural experience; and (5) a very significant financial investment."

SCO explained: "This case is not about the debate about the relative merits of proprietary versus open source software. Nor is this case about IBM right to develop and promote open source software if it decides to do so in furtherance of its independent business objectives, so long as it does so without SCO proprietary information."

Patent search

You can find the reaction of the OSI. A technician may also disagree: SCO, the second-hand buyer of UNIX, did not significantly enhance UNIX. Why investing on something that does not make money? Actually ATT and Novell sold UNIX for the same reason. In early 90 s ATT had a successful Operating System but they did not find the way to convert this technical success into money. Therefore Unix licensees had to substantially invest to port and optimize UNIX performance on their hardware and to support their devices. IBM did this effort twice, one for AIX with the UNIX code, one for Linux with Linux code. A technician would not understand either how a company that releases a Linux distribution (Caldera) can find that Linux contains proprietary code.

So far this story was just another example of weak to strong action in Intellectual Property. However on October 24, 2003 in an answer to IBM SCO wrote: "The GPL violates the US Constitution together with copyright, antitrust and export control laws" (eighth affirmative defense). An article of LinuxWorld reports this news.

Before this answer Darl Mc Bride, the CEO of SCO wrote an open letter to the open source community: "In copyright law, ownership cannot be transferred without express, written authority of a copyright holder. Some have claimed that, because SCO software code was present in software distributed under the GPL, SCO has forfeited its rights to this code. Not so SCO never gave permission, or granted rights, for this to happen. Transfer of copyright ownership without express written authority of all proper parties is null and void. Use of derivative rights in copyrighted material is defined by the scope of a license grant. An authorized derivative work may not be used beyond the scope of a license grant. License grants regarding derivative works vary from license to license some are broad and some are narrow. In other words, the license itself defines the scope of permissive use, and licensees agree to be bound by that definition."

This letter triggered the comment from Grocklaw, another SCO group company: "You have continued to distribute the Linux kernel, despite alleging that it contains infringing source code. Simultaneously, you are attempting to compel purchase of "Linux Intellectual Property" licenses for binary-only use, the terms of which are incompatible with freedoms granted under the GPL. According to the GPL, any violation of its license terms automatically and immediately terminates your permission to modify or distribute the software or derivative works. Note the wording of the GPL:

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

Releasing software under the GPL is not the same as releasing it into the public domain. Authors retain their copyrights to software licensed under the GPL. Even when authors assign their copyrights to someone else, such as to the Free Software Foundation, the copyrights remain valid, but with the new owner. Therefore,

Patent search

subsequent to termination of your permissions under the GPL, you are in the unhappy position of violating the copyrights of the software authors, if you continue to distribute their software. Under copyright law, you are not allowed to distribute at all without their permission — and they have chosen to grant that permission only by means of the GPL."

In its answer to IBM SCO does not give details about its assertion that GPL infringes the Copyright law. LinuxWorld believes that the argument is that under the GPL, mere human beings – American citizens – arrogate to themselves the right to give permission to copy, modify, or redistribute intellectual property, while that kind of regulation of copyright, according to SCO's contention, can and must only be done – per Article 1 Section 8 of the US copyright law – by Congress itself. This explanation is consistent with the open letter of Darl Mc Bride.

The US copyright law (circular 92) says: "unless authorized by the owner of copyright in a computer program any person in possession of a particular copy of a computer program may, for the purposes of direct or indirect commercial advantage, dispose of, or authorize the disposal of that computer program by rental, lease, or lending, or by any other act or practice in the nature of rental, lease, or lending" and "Any exact copies prepared in accordance with the provisions of this section may be leased, sold, or otherwise transferred, along with the copy from which such copies were prepared, only as part of the lease, sale, or other transfer of all rights in the program. Adaptations so prepared may be transferred only with the authorization of the copyright owner." Our understanding is that authorizations granted by the GPL license are well described in the Copyright law and that the issue rose by SCO is whether implicit authorization is legal or not. From our point this is legal because, for instance, opening a CDROM seal is already considered as an implicit approval of license terms. However we think that this is time to clarify this issue in a judgement.

Intellectual Property tasks

We have seen above that Intellectual Property can no longer be ignored. We may also add that this is no longer only the job of specialists. In that respect Intellectual Property follows the evolution of the Society. Since the beginning of the 60 s people are massively educated. Up to the 80 s computers were black boxes operated by aliens talking a language that nobody else understood. Now the whole workforce has a coarse understanding on computer matters and everyone talk the computing jargon. We wrote Air transport because we wanted to illustrate the close integration of all parties involved in providing a service, companies of different size and government agencies, which is, we believe, one of the deepest changes in the Society and because we were asked to. Now people also understand what revenue management and landing slots are. We see that not–so–easy–to–understand concepts of the Air Transport and of the banking system are commonly understood. Intellectual property will become a part of our common culture. Now managers understand that in the next decade they will spend a third of their time on Intellectual property issues. In this section we explain how this time should be spent:

- Intellectual Property out: the information disclosed by a company or organization,
- Intellectual Property in: the information a company should collect.

Intellectual Property out

A company or organization communicates with the rest of the world and therefore constantly discloses information about its technical choices, methods and processes through mails, messages posted on public forums, Web sites, press releases, presentations, documentation, trainings, RFP, products and patents. Employees of large firms frequently get more and better data about their company on Internet than using the firm internal communication systems. This is not always a drawback. The real issue is that Murphy's Law also applies here: things that you would like to keep hidden quickly spread whereas things that you would like to spread are unnoticed.

Our experience is that mails and messages are less important:

1. Mails and messages frequently make sense years after their writing when a new product is released. A prototype can generate a large number of messages and be abandoned later.
2. Mails are usually about technical issues like errors, bugs, coding and administration problems.
3. Authors are usually hard to identify.

Very early in a project a company should take a final decision about its Intellectual Property policy regarding this project:

1. Secrecy
2. Patent protection
3. Publication
4. Open source

Procrastination is not an option because the date when a publication was made or a patent was filed determines who legally invented a process or method. When you take your time to decide you have to keep your innovations secret during the while to keep the choice open. If you opt for publication or for open source sooner you disclose better it is.

When you chose secrecy or patent protection you have to handle three sorts of disclosure:

1. Disclosure to customers such as documentation, trainings and products
2. Disclosure to prospects such as RFP answers and presentations and to partners
3. Disclosure to public such as Web site, press releases, brochures and some presentations (for instance in conferences)

We take the example of a software publisher.

Customers

The license agreement should include a confidentiality clause.

If you sell service with the product the service contract should also include a confidentiality clause like "Each party may disclose confidential information to the other party. The parties shall preserve of the confidentiality

Patent search

of such information during the performance of the service and during sometime [to be defined] as from the Agreement expiry."

Documentation manuals and training support documents should include a page with something like: "This documentation may not be copied, transferred, reproduced, disclosed or duplicated in whole or in part, without the prior consent of [your company]. This documentation is proprietary information of [your company] and protected by the applicable law of [your country] and international treaties." This page should refer to the license agreement. It may also refer to a restricted rights definition. The industry commonly follows the definition used by government for its own purchases. Magic keywords to find samples on the Web are 48 CFR Section 12.212 and 48 CFR 227.7202. 48 stands for Federal Acquisition Regulations System, 12 for Acquisition of commercial items, 212 for Computer software, 227 for Patents, data, and copyrights, 7202 for Statutes pertaining to administrative claims of infringement. You may find these Federal regulations on <http://www.access.gpo.gov/nara/cfr/cfr-table-search.html>.

A secret shared by thousand or more customer employees is longer a secret (a user may copy a product method if this method is not patented and competitors have your documentation) but if you follow the guidelines above this is still a legal secret: you still can file a patent and a competitor cannot show your user manual as evidence in an infringement prosecution or reference your manual in its patent applications.

Prospects and partners

You keep your intellectual property secret though a non-disclosure agreement (NDA). An NDA is a contract whereby the signer agrees not to disclose certain information, except under terms as described in the contract. In a one-way NDA one party agrees to give to a second party confidential information and the second party agrees not to share this information with anyone else for a specified period of time. In a two-way NDA both parties agree to give confidential information and to not share this information with anyone else for a specified period of time.

A one-way NDA may be fine for the first presentation to a prospect. If they have to explain how they intent to use the product, then the prospect will probably ask for a two-way NDA. In case of answer to an RFP or of partnership you also need a two-way NDA. The problem is that each party usually has its own NDA. The stronger party and when parties are of similar strength the customer tends to impose its NDA or to refuse to sign the NDA of the weaker party. This is especially true in countries and businesses in which actors are not fully aware of Intellectual Property issues. NDA negotiation is an extra burden for an already tough job. Furthermore when a sale or an RFP cycle is three to six months, when a partnership lasts two years, the consequence of negligence in an NDA may be fully appreciated eight or ten years later, which is more than the average experience of the actors in their job and sometimes more than the time they will work for their company.

You can find examples of NDA at <http://www.uiausa.com/Nondisclosure.htm> and <http://www.uiausa.com/TutleNDA.htm>.

Public

Here we consider Web site, press releases, brochures and public presentations. Intellectually property is made of functions whose combination solves a set of problems and of how these functions are implemented. Look at the description part and at the drawings of a software patent. You will find screenshots, block diagrams and relatively high-level explanations. The description of a patent contains the same data as an article.

A site that does not disclose intellectual property is a site that does not answer to the how question. Such site does not attract visitors because it contains no information and is poorly indexed by search engines because it has no content. If you choose secrecy or patenting you cannot effectively use one of the cheapest and most effective communication means. You may however design a site with a public part and a private, password protected, part. When you can identify the users you are in the same situation as for paper documentation: the content is not public but at the same time the very last persons you would like to see on your site will be your most regular visitors. You may allow self-registration. In that case ask the user to acknowledge a confidentiality clause. You can check her identity by sending a mail with a confirmation number. Oracle uses such a system. You can find in Oracle documents: "The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited." Only highly motivated users accept to register, which is both a drawback (few visitors) and a benefit (good quality visitors).

A press release is a news exercise and because of that an unlikely place for intellectual property disclosure. Neither the format nor the reader expectations, nor the press release goal really enable disclosures. Furthermore in a company that has software intellectual property people who create and understand this intellectual property are in development teams and not a communication services. We only mention press releases because we were asked if a press release could prevent patenting. To check if a press release contains intellectual property answer the following questions:

- Does it explain how something is working?
- Does it contain diagrams?
- Does it contain screenshots and explanations?

If you answer no to all of these questions then do not worry about the press release. This also works for brochures. A public technical presentation is a much more dangerous exercise for two reasons:

- People attend a presentation to learn something
- The objective of the presentation is to show that the company is skilled and experienced but to achieve this objective the speaker needs to have some matter. Less the speaker is experienced in the difficult art of communication more she needs to disclose and she is frequently inexperienced. This is another illustration of the type 3 work of Society and Computing. Press release writing and brochureware are comfortable jobs. There is no shortage of readers, suggestions and enhancements for these documents. Presentation is hard work implying a strong involvement of the speaker. Therefore technical presentations are frequently delegated to junior programmers who are not even given the time to properly prepare their performance.

Conclusion

This is difficult to not disclose intellectual property. In an ideal world the company would define what it is ready to disclose and record what and to whom that it has disclosed. Things made public still have a value. We can illustrate it with a simple example. Your competitor patents a method. If you can show that you owned the method before, you still can use the method but this is bad for your business and for your customers. If you can prove that you presented the method to the public the patent will be rejected.

Patenting has the merit of forcing people to think about these issues. Most of the time what we call secrecy should be called "not care". Open Sourcing your development implies taking care to not infringe copyright of others. Publishing requires extra work. Secret is definitely the cheapest option. For instance it makes sense to keep internal developments for internal users secret. Even the most dedicated patent portfolio makers realize that not everything can be patented: there is always novelty in a large project but this novelty may have a so narrow scope that this is not worth to protect it. Large firms use all methods:

- Patent portfolios
- Open Source
- Publication
- Secrecy

The most obvious example is IBM, which has the biggest patent portfolio and is also one of the biggest Open Source publishers. Companies like BEA, Cisco and Oracle publish large amount of content on their Web sites. Even a small company probably has to make some things public in a conscious way.

Intellectual Property in

Intellectual Property in is obviously related to competitive analysis and market intelligence. However whereas the latter aims to answer the questions: Who are our competitors? How well are they doing? What are their plans? Intellectual Property in aims to answer a single question: Are we allowed to do that?

In some way competitive analysis and market intelligence are the other side of Intellectual Property out and suffers the same problem. Marketing, sales forces and everybody who meet customers is doing market intelligence. A company collects a huge amount of market data but these data are disseminated. Furthermore it is difficult to measure the relative importance of these data, which are in different formats, more or less reliable and hard to link together. However despite their flaws competitive analysis and market intelligence are the first input of Intellectual Property in. They provide raw data such as competitor names, market and mind shares.

Understanding the "that" in "are we allowed to do that?" is another difficulty. Even if you do not plan to patent you must follow the principle exposed above in Patent writing: the product design is the other input of Intellectual Property in. You use this input to write a reference text, which is a patent subset:

- The field of the product
- Background information: what people will need to understand the product
- Prior art: Problems faced in the product area and existing solutions. Briefly describe how these

Patent search

solutions are working and the problems that they do not solve.

- Summary of the product design: How the product solves one or several of these problems

Use the reference text to make searches in the patent databases with the toolkit presented above. You can use the patent class of the found patents to broaden the search scope. Look at the referenced patents. Look at the patents referencing the patents. At this stage you may have found forty or more patents. Then consolidate your results with the list of competitors identified by the competitive analysis and market intelligence. A direct competitor will go to court with a case weaker than a company that is on a different market or that is also your provider, customer or partner. It has also more chance to be granted an injunction, in which case you will have to stop selling or using your product. On the other hand patent search may identify new entrants on your market. In that respect Intellectual Property in also feeds competitive analysis and market intelligence. You must also consider patent families: a patent filed in a country may have been promoted to other countries through PCT, which implies a stronger commitment of the patentee. Such patent may represent a stronger threat for your product.

Next compare the independent claims of each remaining patent with the product design. If you find a match between an independent claim and a part of the design ask for a redesign. You should also send a document presenting the most threatening patents and claims to analysts and programmers. Then throughout the project you should periodically check for new patent applications. This is a tough and time-consuming job. If you frequently file patents and if you never stop being sued (which means that you make money and have an enviable position on your market) you represent a sizeable and regular source of revenue for your lawyers and it makes sense to ask their help. A software patent attorney usually knows something about software, which is not always the case of legal departments. On the other hand we do not recommend hiring a patent attorney only to assess legal risks related to a particular project:

- The attorney must understand the activity of the company and know its market and competitors
- The attorney must check for new patent applications throughout the project development
- This is a time-consuming task that highly depends on the project. An assessment may require ten to hundred and more hours of research. Such assessment cannot be cheap. Furthermore the customer has no means to check how extensive and thorough the research was.

Therefore it can make sense to do the search yourself. This is a senior programmer or designer job that requires a holistic view of the company and market. This document may help starting in this activity. Public databases and our toolkit are probably enough to identify threats and find corrective actions. You can also use <http://www.freepatentsonline.com/> as an alternative to the USPTO web site. freepatentsonline allows downloading patent files in PDF format, which may be preferred notably for printing. Today freepatentsonline only lists granted patents. In the future they should also list patent applications. You may need to check other sources of information. The Patent Information Users Group site may be a good starting point: "This Patent Information Users Group, Inc. (PIUG), the International Society for Patent Information is a not-for-profit organization for individuals having a professional, scientific or technical interest in patent information." This site contains a good FAQ. You can also use commercial databases and tools. You can find a list of vendors at <http://www.piug.org/vendor.html>. There is a good presentation of patent tools, Software Tools for Analyzing Patents by Anthony Trippe. The most popular commercial sites may be:

- qpat
- micropatent

Patent search

- [IBM patent server](#)
- [Derwent patent explorer](#)
- <http://www.ip.com>

Corporate subscriptions can be expensive but these sites frequently offer pay-per-use service (vouchers, one day subscription) at a much lower price.

You can be in three situations:

1. The current design or a feature of the product literally infringes the claims of a patent or a patent application, which is likely to be granted with its current scope. In this case the project management has to (1) either modify the product to no longer infringe the patent claims (2) cancel the project (3) or negotiate a license agreement with the patentee.
2. You identified patents and patent applications belonging to the product domain and filed by companies that compete or may compete with your company. You found that none of these patents and patent application claims the product design and features. Then the project development and releasing can go on.
3. You are in the twilight zone. The Federal Circuit explained that "'willfulness' in infringement, as in life, is not an all-or-nothing trait, but one of degree. The law recognizes that infringement may range from unknowing, or accidental, to deliberate, or reckless, disregard of a patentee's legal rights. [...] The term 'willfulness' thus reflects a threshold of culpability in the act of infringement. [...] The law of willful infringement does not search for minimally tolerable behavior, but requires prudent, and ethical, legal and commercial actions. Thus precedent displays the consistent theme of whether a prudent person would have had sound reason to believe that the patent was not infringed or was invalid or unenforceable, and would be so held if litigated." You may find that the product implements neither the limitations of the patent claims nor any reasonable equivalent to these claims limitations. You may also find prior art that make the patent not novel or obvious and therefore invalid. But you recognize that the patentee and a court can find that the product infringes the patent. In this case ask for an opinion of counsel as explained in [Software and business method patents](#). If the opinion is positive go on with the project development and releasing as in 2. Otherwise proceed as exposed in 1.

You have to focus on competitors patents and patent applications because:

1. Checking all relevant patents and patent applications would require an unreasonable amount of work.
2. A patentee who is not a competitor does not sell less because of the infringement and a court is unlikely to issue an injunction forbidding you to sell or use the product. The Federal Circuit has "repeatedly endorsed the conceptual framework of a hypothetical negotiation between patentee and infringer as a mean for determining a reasonable royalty" to determine what the infringer has to pay to the patent holder. So you are less exposed and the damage awarded to the patentee is related (though usually much higher) to the license fee you would have paid if you had negotiated with the patentee.
3. You are supposed to know what your competitors are doing and the patents they own. This due diligence does not extend to other businesses.

The net result is that you will be sometimes notified by a patentee about a patent infringement because:

Patent search

1. The patent is in the twilight zone.
2. You gave a relevant patent a miss because this patent was filed by a company or individual not known in your business.
3. You simply did not find a relevant patent. The search process is not 100% reliable.

There are several reasons to not go to court:

1. litigations are expensive and time-consuming; if you are in a profitable and growing business it is usually better to spend your time on your business rather than on litigations;
2. courts presume that patents are valid and the burden of proof is on the defendant (you);
3. in USA patent infringement cases are generally ruled by juries (choice of the plaintiff – the patentee) and the experience shows that juries' decisions and awarded damages are hard to predict.

Uncertainty and litigation costs are also issues for the patentee. And if he is awarded damages these damages will be awarded after a couple of years. So the interest of the patentee is also to negotiate a license agreement.

There are two factors to consider when you prepare the negotiation:

1. The value of the patent process or method for your business. I discuss this issue in the Damage background section of Software and business method patents. You have to answer three questions (1) can you design around the patent and at which cost? (2) what is the cost for you and your customers if you disable the functions that allegedly infringe the patent? (3) how much money are you making thank to the patent process or method?
2. The likeliness of patentee success in court and the likeliness of patent invalidation in a patent re-examination. Here you have to identify prior art that was not considered or improperly presented in the patent examination.

A reader informed me about a Settlement help site containing a "large Directory of Settlements Help Sites, financial websites and general sites of interest". An article of this site entitled "How \$50 Million Settlement Was Reached Hours Before Trial Began– Esquire Litigation Solutions Seen As Key In Forcing Settlement" shows the way you should present your data. In the negotiation you must demonstrate that:

1. the patent could be found not novel or obvious and invalidated;
2. you can design-around the patent or live without using the protected system or method.