

The Case For Patent Intelligence

Thursday, May 18, 2006 --- The lights are shining in the 'surgery.' A man in a lab coat is operating, but we cannot quite see on what. But, by means of his precise and delicate exploratory surgery, he will learn exactly what he needs to know. The expert described is not a doctor, but an analyst using reverse engineering to determine how a cell phone works and if it might be infringing any patents.

Those unfamiliar with reverse engineering sometimes associate it with hackers, copyright violation and theft. Nothing could be further from the truth. Reverse engineering is a means of maintaining competition that is fair and healthy for the marketplace.

Reverse engineering is an efficient and effective way for companies to protect patents and ensure that competitors are not using them illegally. It also helps companies protect themselves by ensuring they do not infringe on their competitors' patents.

Nearly every large corporation uses reverse engineering as a tool for competitive analysis and as a means to uncover possible patent infringements.

* A Means to Monitor Patents *

Today, despite extreme competition, companies demand the maintenance of a fair and level playing field. One way to guarantee fairness among competitors is for all parties to have equal access to patent intelligence through the use of reverse engineering.

The world's most innovative and creative companies guard their intellectual property by acquiring patents to protect their inventions. Obtaining the patent, however, is only the first step. Companies must aggressively monitor industry and competitive products in order to ensure that no one is copying their innovations. The most effective way to monitor patents is to reverse engineer competing products to gain patent intelligence.

Reverse engineering is a widely used, legally accepted way to gain competitive information. Just as making a copy of a CD for back-up purposes is 'fair use' under copyright law, reverse engineering for certain purposes is also considered 'fair use.'

* Courts Support *

Using the 'fair use' argument from copyright law, a clear line of cases support

reverse engineering of software for the purposes of understanding functionality and for developing a different version of information or code. As long as a device or piece of software has been legitimately obtained, reverse engineering for the purposes of investigating and understanding the program is considered a legitimate purpose even if carried out in a commercial context.

The jurisprudential support for reverse engineering can be reviewed in three prominent cases.

1) *Sega v. Accolade*

The leading case on 'fair use' in the context of the reverse engineering of computer software is *Sega Enterprises Ltd. v. Accolade Inc.* 977 F.2d 1510 (9th Cir. 1992) ("Sega") decided by the Ninth Circuit Court of Appeal. The defendant, Accolade, had analyzed Sega's video game programs in order to determine the Genesis console compatibility requirements. Sega challenged Accolade's reverse engineering process on the grounds that the intermediate copying during disassembly constituted a violation of Sega's copyright in the game cartridge object code.

The Court held that Accolade's intermediate copying of the Sega object code would have constituted copyright infringement but was protected by the 'fair use' exception. The Court considered four relevant factors:

a) The purpose and character of the use

In considering the purpose of the use, the court determined that even though Accolade eventually used the information obtained by the copying to produce a competing product, this was not conclusive. Although the end goal was commercial, the purpose was to study the functional requirements only, and therefore this use was legitimate.

b) The nature of the work

The court noted that not all copyrighted works are entitled to the same level of protection. In particular, the protection provided by copyright does not extend to the ideas underlying the work. The fact that computer programs couldn't be examined without a certain amount of literal copying proved to be a significant factor.

c) The amount of the work copied

In considering the amount of the work copied, the court noted that even though Accolade copied the entire work, it would still not by definition preclude a finding of 'fair use.'

d) The effect of the use upon the potential market for the copyrighted work

The court cautioned against making this factor determinative. The court conceded that Accolade's entry into the game market undoubtedly affected the market for Genesis-compatible games in an indirect fashion, but further indicated that allowing Sega to try to monopolize the market by making it impossible for others to compete ran counter to the statutory purpose of the

Copyright Act to promote creative expression. The court therefore found the fourth factor favored Accolade despite the minor economic loss that Sega might suffer.

Therefore, the court found that where disassembly is the only means to gain access to the ideas and functional elements embedded in a copyrighted computer program and where the reason for seeking such access is to achieve interoperability, disassembly is a 'fair use' of the copyrighted work.

2) Atari v. Nintendo

In *Atari Games Corp. v. Nintendo of America, Inc.* 975 F.2d 832 (Fed. Cir. 1992) ("Atari Games"), a case decided shortly before Sega, the Federal Circuit Court reached a similar conclusion concerning the legality of reverse engineering. However, since the issue arose in the context of a preliminary injunction, the analysis was not as detailed as in the Sega case. Further, in *Atari Games*, the fair use issue was complicated by the fact that Atari Games' lawyers lied to the U.S. Copyright Office to obtain the registration copy of Nintendo source code so that the firm's engineers could use it to finalize the development of compatible games. Nevertheless, the Court ruled that the initial de-compilation copying constituted 'fair use.'

3) Sony Computer v. Connectix

The Ninth Circuit Court recently reaffirmed the Sega ruling in *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). The main difference between this and the Sega case was that Connectix disassembled Sony programs in order to develop emulation software to allow owners of Apple iMac computers to play Sony PlayStation games. That is, Connectix reverse-engineered in order to make a competing platform, not to make compatible games. The appellate court perceived no legal difference between the de-compilation-for-interoperability considerations pertinent to development of competing platforms and those pertinent to games.

It appears that the U.S. doctrine of 'fair use' favors a relatively wide scope for reverse engineering. According to current case law, reverse engineering is allowed for the creation of competing hardware platforms and interoperable computer programs.

* Enforcing Software Patents *

Guarding against patent infringement is one key reason for companies to reverse engineer a product. In the electronics hardware space, reverse engineering is not only widespread and acceptable but is in fact the only reliable way to prove infringement at the semiconductor level. Specific provisions that allow reverse engineering were added to the Semiconductor Chip Protection Act at the specific request of electronics companies, because they understand the importance of reverse engineering from the perspective of both patent enforcement as well as competitive intelligence.

Acceptance of reverse engineering of software is not yet entirely entrenched in the legal system. However, it is often just as necessary for software as it is for hardware. If a patent holder believes its patents are being infringed by a competitor's software, then the patent holder should have the right to discover whether that is true. Software reverse engineering allows for that.

On the open market, the patent holder can purchase the product, which includes the code in question. The patent holder would need to extract this code. Sometimes this can be simple (for example, by reading it from disk). In other cases, it can prove somewhat more complex. Numerous techniques exist for code extraction. Software embedded in a hardware product would typically be in a binary format. The analysis team must convert this to a human readable form. In many cases, disassembly to assembler or an intermediate software language is quite simple. At that point, an experienced software engineer can begin analysis of the code.

Fortunately, there are also a multitude of ways to develop evidence of software patent infringement that do not require that it be reverse engineered back to intermediate or source code.

For example, the functionality involved in patent claims can often be documented. One method for documenting functionality is to run targeted tests on a system and monitor the response in order to infer the use of certain code. That is, by checking the functionality of a system under certain conditions, one can sometimes show that specific code must be used. In certain cases, the code might be quite accessible and not in binary format. In other cases, code can be intercepted by hardware monitoring during functional operation, for instance, when downloading a program from Flash memory to a processor when a system is booting up.

Software patents are often quite broad, and any code used to create a specific end result might be an infringement. In such cases, of course, only the end result needs to be shown. Finally, occasionally the code is available as source code, and only the analysis step must be performed.

* Conclusion *

Reverse engineering for legitimate competitive research and the development of new innovations is a long-standing and respected method of gaining competitive information.

Numerous court decisions have upheld the view that using reverse engineering to gain patent intelligence is justified under fair use provisions. Ultimately, reverse engineering is an important means of keeping companies ethical in highly competitive industries and, quite often, the only way to garner crucial patent intelligence.

--By Terry Ludlow, Founder and CEO of Chipworks

Terry Ludlow is CEO and Founder of Chipworks, a company that reverse

engineers and analyzes semiconductor and microelectronic systems for two distinct and complementary groups. Patent Intelligence customers are law firms and Intellectual Property groups who need Chipworks' support to defend or enhance their licensing positions. For information: visit www.chipworks.com, email info@chipworks.com or call 613.829.0414.