

# Patent Talk

## PROTECTING COMPUTER SOFTWARE: PATENT OR COPYRIGHT

D. Lloyd Rice

Lafayette, Colorado 80026

Gregory Aharonian

San Francisco, California 94111

Lloyd Rice, retired software engineer and editor of the Patent Reviews section of the *Journal of the Acoustical Society of America*, discusses the questions “Can computer software be patented?” “Can it be copyrighted?” “Which of these makes more sense?” with Greg Aharonian, San Francisco patent attorney and editor of *PatNews*, a widely read Internet newsletter on the current state of patents and copyrights.



D. Lloyd Rice



Gregory Aharonian

to say that a copyright would be useless as a way to protect a program. The style of how the program is expressed is of little concern in the real task of the software, which is to produce a particular machine behavior.

Greg: I agree. Court decisions on software copyright are generally nonsensical because they keep on

trying to extend copyright to functionality (despite what it says in United States Code Title 17, Section 102b), because with software, expression is secondary.

Lloyd: Patents and copyrights are firmly rooted in the capitalist principle that individuals and business organizations have a right to receive profits from their creativity. The original concept was meant to apply to the results of physical creations, but patents and copyrights extended that right to the results of mental activities. Patents also allow the public disclosure of ideas which would otherwise be trade secrets. In exchange for that disclosure, the inventor gets a limited monopoly on the idea.

Lloyd: Many have argued that software is essentially a mathematical formula. Mathematical expressions are not and have never been patentable. Does this position then apply to software?

Is it true that a patent protects an idea, whereas a copyright protects the way that idea is expressed?

Greg: You get the law wrong. Mathematical expressions by themselves are not patentable. But mathematical expressions in specific applications, with inputs and outputs for a real world problem, are patentable. So as long as the software addresses a real world problem, it is patentable.

Greg: Copyright law does not define the terms “idea” or “expression,” letting judges apply these terms at their arbitrary discretion. Protecting an adaptation via copyright is equivalent to a narrowly defined patent claim. Protecting a specific, artistic article of manufacture via patent is equivalent to a copyright, but is more powerful. Lawyers have completely ignored the overlap of these two systems.

Lloyd: It can be argued, based on some social philosophies, that any mental creations should be public property. Taking the view that software should not be protectable at all as intellectual property, a culture has grown up around the idea that software is a public resource and that it should be freely available to all who want to read, understand, or modify it for their own purposes or for further distribution. This point of view grew up with the advent of small computers in the hands of individuals. It took root with the formation of the Free Software Foundation (FSF) by Richard Stallman and came into its own right as an acceptable form of software distribution with the growth of the Linux operating system, written by Linus Torvalds.

Lloyd: How does this apply to software? The purpose of software is to make a machine behave in a particular way. But many different forms of the software could result in exactly the same machine behavior. For example, in the computer languages C or C++, the two statements

```
for ( ; ; ) { xxx }  
and  
while ( 1 ) { xxx }
```

mean precisely the same thing and would be completely interchangeable in any program which used one or the other. Nearly every statement in a program will have many such interchangeable forms of expression. This fact clearly seems

Greg: Government and university laboratories were developing and making available tons of public domain software many years before the FSF. And that software was truly free, without the absurd conditions imposed by Stallman's General Public License (GPL).

Lloyd: One of the reasons the purveyors of open source software want to stick with copyright instead of going with

patents is the loose (or nonexistent) definition of derivative works. But with patents, it all depends upon how the judge chooses to view the claims.

Greg: Copyright law is much worse. Not until a copyright trial, when a judge (or his expert) does an *abstractioning*, where he creates the equivalent of claims, do you have any notice of what the boundaries are. I have a right to know which of the ideas in the source code I can freely use. At least with patents, you get to see the claims before trial, when the patent issues. But with copyright, you are at the whims of a judge untrained in software to create the “claims” while you are being sued.

So if patent litigation is bad, copyright litigation is worse because a judge during trial first has to create the claims before the equivalent of a patent lawsuit occurs. For more on copyright abstractioning, examine Google for the Abstraction-Filtration-Comparison test from a case titled *Computer Associates v. Altai*.

Open source proponents choose copyright because that is what Stallman chose for GPL in the mid 1980s. At the

time, protecting software via patents and copyright was still in flux as the courts figured out what to do. People like Stallman and the Patent Office hoped the courts would negate software patents and shift protection over to copyright. The courts did the exact opposite; greatly expanding software patenting and greatly limiting copyright (*Lotus v. Borland*, for example).

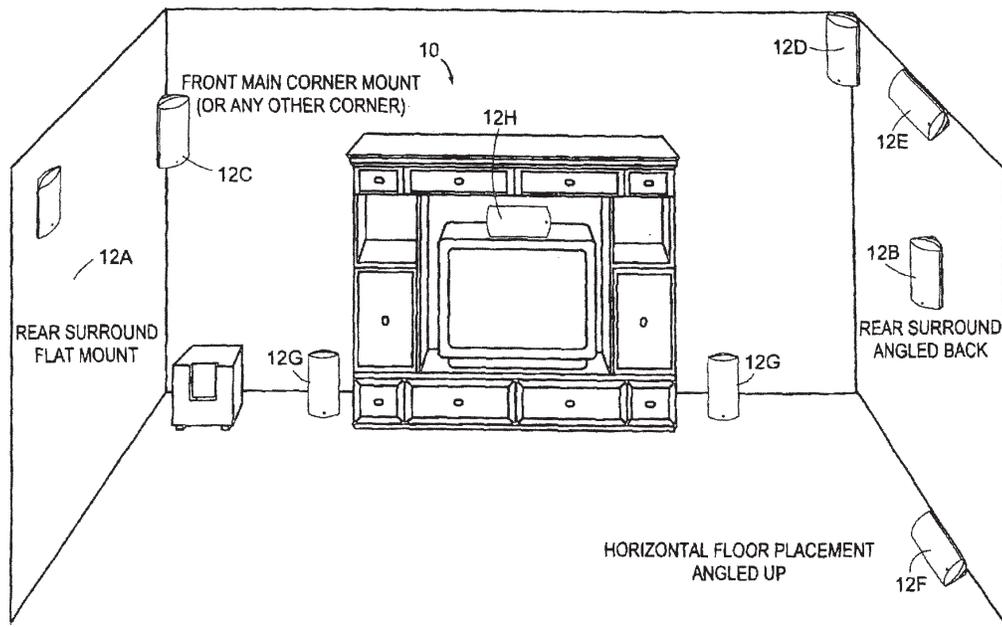
Thus, questions like “what is copyrightable in a source code,” “what is a derivative,” remain completely unanswered, partly because many are afraid that the answer in each case is “not much” and “who knows.” For GPL to rest on copyright is sheer lunacy—a good patent litigator could *shred* the license in a court case.**AT**

Editor’s Note—The opinions expressed in this article are those of the authors and do not necessarily reflect those of the Editor, *Acoustics Today*, or the Acoustical Society of America.

6,845,840

### 43.38.Tj SURFACE MOUNTED LOUDSPEAKER AND BRACKET FOR THE MOUNTING THEREOF

Jeffrey N. Cowan *et al.*, assignors to Boston Acoustics, Incorporated  
25 January 2005 (Class 181/150); filed 9 August 2001



A means is described of attaching a loudspeaker enclosure of a certain form in many different orientations while also providing “good aesthetics.” Question: how many domestic divas would concur with the adjective “good” as applied in this context?—NAS