SOFTWARE PROTECTION: COPYRIGHTS, PATENTS, TRADE SECRETS AND/OR *SUI GENERIS*

Karl F. Jorda

What is the best form of protection for software has been and still is a most unsettled and vexing — and hence very topical — issue in intellectual property (IP) law and practice.

Congress, of course, did amend our copyright law in 1980 to make it clear that software is copyrightable.[1] Likewise, legislation was enacted in foreign countries and the European Union in past years, stipulating that software is only copyrightable, i.e. not patentable. Trade Related Intellectual Property (TRIP) also requires that copyright protection be provided by World Trade Organization countries. Thus, it is not surprising that Ralph Oman, the former Register of Copyrights, and others maintain that an international consensus in favor of copyright protection has emerged, even though many believe that copyright protection is an *artificial construct* inasmuch as the aims of copyright law and computer programming are diametrically opposed, the former stressing subjective, individualistic, creative elements, and the latter, objective, technical and scientific systematization. Software is functional, non-literal by nature as it performs a task or generates an output.

Thus, there are many authors and practitioners here and abroad who believe that copyright laws are inappropriate as forms of protection and it is patent law and/or *sui generis* systems which would offer better protection for software. And more and more countries follow the lead of the United States and sanction the patenting of software. Headlines of recent articles bear this out; to wit *The Case for Software Patent Protection*; *Software Patents Come of Age*; and *Patents, Not Copyright, Poised for Bigger Byte of Software.* But there are significant problems with software patents as illustrated by the following titles: *Now You See It, Now You Don't: Was It a Patentable Machine or an Unpatentable 'Algorithm?'*; *Software Patent Protection: Debugging the Current System*, etc. According to Professor Hollaar (University of Utah) it is high time that Congress "clarify the patentability of software-based inventions."[2]

[1] 17 U.S.C. § 117 (1994).

[2] *See* Hollaar, *Justice Douglas Was Right: The Need for Congressional Action on Software Patents*, 24 AIPLA Q. J. 283, 305 1996.

The shift to patents is also influenced by the recent decisional trend limiting the scope of copyright protection on the one hand, while expanding the scope of patent protection for software on the other hand.

With software protection being a practitioner's nightmare (as one article bemoans) and with both patent and copyright forms of protection being *Procrustean Beds*, it shouldn't come as a surprise that the notion of a *sui generis* form of protection for software, in lieu of or in addition to present routes of protection, has considerable appeal.

Professor Samuelson's 1994 *Manifesto Concerning the Legal Protection of Computer Programs*[3] comes immediately to mind as well as Richard Stern's *sui generis* utility model law proposal, launched in 1993.[4] Indeed, I remember well that the first impulse by the IP profession back in 1965 when the issue first arose, was to provide a *sui generis* form of protection, as was fashioned (improvidently according to some practitioners) in 1984 for semiconductor chips via the Semiconductor Chip Protection Act[5] and will likely be done in the near future for databases.

[3] SAMUELSON, MANIFESTO CONCERNING THE LEGAL PROTECTION OF COMPUTER PROGRAMS (1994).

[4] *See generally* Richard Stern, *sui generis* utility model law proposal (1993).

[5] SEMICONDUCTOR CHIP PROTECTION ACT of 1984, 17 U.S.C. § 901 (1994).