Inventing Around Copyright

Dan Burk

Professor, University of California, Irvine School of Law Irvine, CA

Patent law has long harbored the concept of "inventing around," under which competitors to a patent holder may be expected, and even encouraged, to design their technologies so as to skirt the boundaries defined by patent claims. Inventing around is sometimes viewed as a positive effect of patent rights. The Federal Circuit in particular has touted inventing around as a spur to innovation, suggesting that rights which might be viewed as impeding competitors actually force competitors to become more innovative in the course of avoiding infringement. Other commentators have been less enthusiastic about the concept, observing that inventing around patent rights may well lead to inefficient and duplicative invention, prompting development of unneeded or second best alternatives to patented technologies. It has become increasingly clear that, for better or for worse, copyright also fosters inventing around. Copyright is not based on written claims, but because copyright links exclusive rights to technological actions such as reproduction, distribution, or transmission, the language of the copyright statute, and judicial readings of the statute, create boundaries around which potential infringers may technologically navigate. Thus, for example, peer-to-peer file networks after Napster intentionally adopted architectures that would avoid the basis for secondary liability found by the Ninth Circuit in the Napster litigation. Similarly, the Aereo technology currently before the Supreme Court was explicitly designed to conform to non-infringing definitions of private transmission found in past court decisions. But in copyright, unlike patent, there has been little analysis of the tendency to foster alternative technological development. In this paper I draw upon previous analyses of inventing around in patent law to assess the benefits and detriments of inventing around in copyright.

Email: dburk@law.uci.edu