From: Ken Stults [e-mail redacted]
Sent: Monday, September 27, 2010 3:30 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: No Software Patents

I am not a patent attorney, and my company owns no patents, yet I have learned a great deal about patent law over the last few years the hard (and expensive) way – by being sued for infringement of a software patent. Because the 'protections' of our legal system come *only* at very high cost, the decisions of government that expose citizens to legal attack and increase the likelihood of their need for legal protection should not be taken lightly. Software patents are an excellent example of government action that increases exposure of its citizens to the ridiculously high cost of the legal system with virtually no counterbalancing benefit to society.

The patent system exists on the premise that I will not bother to invest my time in creating a new kind of thing, even if it promises to have great market potential and benefit to society, if predatory competitors are free to steal my design and benefit from my creativity at my expense. The theory is, if I understand it correctly, that the protections of the patent system encourage inventiveness among the entire population, regardless of their ability to quickly and efficiently take their ideas to market, and thus reap greater benefits to society. This theory may well be valid for inventions of physical objects that can be described and drawn in explicit detail, assembled and disassembled, and whose structure and purpose can generally be readily differentiated from the structure and purpose of another physical object of similar or different design. The theory does not hold up for software patents; in fact it fails so completely that the patent system applied to software works against the public interest by restricting availability of 'innovative' solutions, adds cost without commensurate benefit, and plays no role in encouraging innovation.

The core problem is the very nature of software innovation itself. Software is useless without a computer, and computers are useless without software. Yet together the possibilities are endless, usually in such a seamlessly evolutionary way that it is impossible to 'hold' a given software application as being separate and distinct from what went before it any more than it is possible to 'hold' any product of any evolutionary process as separate and distinct from what went before it. The main difference between the software evolutionary process and biological evolutionary processes, however, is speed, which only compounds the problem of subjecting this process to a necessarily slow patent system. The patent used to extract nearly a million dollars from my small company, and a few more millions from a similar small company, is an excellent case-in-point.

Computers led to the creation of digital databases; digital databases led to the evolution of relational databases, where data points common to separate databases (e.g. name, address) allowed information on those separate databases to be cross-referenced, or 'integrated'. Relational databases, in turn, have evolved profusely such that our society in this information age is literally built on a foundation of interconnected information.   We expect a single customer service representative of a large, complex entity such as the phone company, for example, to know who we are, where we live, what services we have subscribed to, the status of our accounts, and our service history even if this information is collectively stored in multiple discrete databases, usually in different physical locations. We become quite frustrated if the customer service representative cannot or will not connect all these dots, and very rapidly.

Yet a patent was issued for a software application that applied this ubiquitous concept to the very narrow field of helicopter ambulance transport. The fact that linking common information among disparate databases was already well established throughout the economy, even as close as the ground ambulance software market, was apparently not a factor. The patent office believed (erroneously) that this 'invention' represented the first application of the concept to the very specific transport of patients by helicopter, and thus deserved 21 years of market protection!

If this had been a physical device that could be held, examined, assembled, disassembled, described and schematically drawn, it would have been crystal clear how similar, if not identical, it was to previously existing physical devices designed to accomplish the same purpose, or at least how obvious the minor modifications were. Being software, however, the invention was merely someone using existing tools (a computer and relational database application development software) to create exactly the kind of thing those tools were designed to help create. But none of that – the 'stuff it was built from/on, the underlying structure and architecture that <u>the present inventors did not invent but merely used as intended</u> to take one more tiny and obvious evolutionary step – was visible or apparent to the patent office. The described software was considered a 'thing', and its schematic drawings of linked information processes were considered to be the prescribed assemblage of 'parts'.

Since it is difficult to imagine a physical device performing the same function and solving the same problem in entirely different applications without modification, it is easy to see why looking at a software 'invention' in the same way one would look at machine invention can lead to problems. In fact, since software and computers deal in the realm of information, but generally without particular concern for exactly what type of information, once the problem of interlinking common information in disparate databases was solved once, it was solved for all conceivable applications. There is absolutely nothing impressively innovative in applying the same information management to helicopter transport as to ground ambulance transport as to telephone company customer service. The tools are all there ready for use in any conceivable arena.

To me, the invention was the computer/software complex, which is irreducibly complex in that either component makes no sense without the other. This invention has sparked a rapid paced evolutionary process completely inconsistent with defining the metes and bounds of any one software application, each application being either essentially the same as pre-existing applications but merely applied to different information sets, or the natural and logical evolutionary embellishments of pre-existing applications. Software will never create a new 'machine' merely by running inside of it, and it will not even fundamentally transform it. The 'machine' will still be an information processor and it will process information as instructed by its software, without which it has no function, meaning or purpose. New software does not create a new invention, it merely confirms how remarkable the invention of which it has always been an integral part.

My company builds innovative software, but we would never consider applying for a patent. If we did, the sole reason would be to <u>slow our competitors down and increase their cost of doing</u>

business. The incentive for software innovation is the fast paced information market itself, and the very low market entry barriers compared with manufacturing and distributing physical things. Patent protection offers not a whit of additional incentive to create, and by the time a uselessly vague software patent is issued the evolutionary state has almost always moved beyond it, rendering the process just as useless for getting valuable and otherwise inaccessible technical information into the public domain. What other use does the patent system have? The answer is the unintended but very powerful use of slowing the software evolutionary process our society has grown to depend on, and adding enormously to the cost of doing business in the information market. If the goals of the patent system are to protect and reward innovators and speed the process of getting valuable information and tools into the public domain, stop issuing software patents and both goals will be admirably achieved, with the side benefit of lower cost and perhaps even more jobs.

Health
Services
Integration

Kenneth Stults, CEO
(707) 571-2322 office
(707) 696-2283 cell
krstults@hsihealth.com