

# Disclosure

Ben Klemens

27 February 2008

One of the oft-cited stories behind patents is that they are a contract for disclosure. The inventor could have kept her<sup>1</sup> invention a secret, but the government offers her a deal that if she reveals her technology so others can learn from it, then the government will give her a limited monopoly for twenty years (from the date of application).

This is not a fundamental reason for patent law. It's icing. The fundamental reason is the ex-post research grant I'd mentioned earlier.

Which is a good thing, because the disclosure story really only works in theory.

Those of you in technical fields can verify this through introspection: when you last worked on a project, did you first check the journals or the Patent Office? Did it even occur to you to learn from patents? You can search sites of technical working papers, like arxiv.org, for patent references, and you'll find a handful, but a handful of references out of hundreds of thousands of patents is not very impressive. If the patent record were a journal, the journal's impact factor would be right around zero.<sup>2</sup>

Firms in all industries tend not to search patent databases for technological instruction. Arora et al. [2003] state that "patent disclosures appeared to have no measurable impact on information flows from other firms, and therefore no measurable effect on R&D productivity." [p 17] Arundel [2001] finds that "a consistent result in survey research on the use of patent databases is that they are among the least important external information sources available to firms."

**Software: even less disclosure** Campbell-Kelly [2005] wrote a paper whose title typifies pro-software patent research: "Not All Bad: An Historical Perspective on Software Patents." His primary argument for why software patents aren't all bad is that patents set a disclosure requirement.

But disclosure is even less functional in software than for general industries.

For example, Campbell-Kelly cites the algorithm for LZW (Lempel-Ziv-Welch) encoding as a success story for patents. However, as Campbell-Kelly notes, Terry Welch had published the algorithm in a peer-reviewed journal within a few months of applying for the patent, but years before receiving it [Welch, 1984]. If Mr Welch

---

<sup>1</sup>Following a suggestion by Thomson [2001], I have chosen the gender of representative agents in this book by flipping a coin.

<sup>2</sup>Now, patents themselves cite prior patents all the time. Patent attorneys, as opposed to inventors, are no doubt learning up a storm from the patents they are reading all day long. But to say that patents are intended to educate attorneys writing other patents is ingrown logic, to say the least. There's still that disconnect between the people writing and learning from patents and the people who are at work developing new technology.

had truly wanted to keep the LZW algorithm a trade secret should patent protection be unavailable, then he would have waited for the ink on his patent application to dry before publishing the algorithm.

In *Northern Telecom v. Datapoint*,<sup>3</sup> the CAFC ruled that writing source code—in fact, authoring anything more detailed than the broad flowchart describing the overall logic of the design—is a “mere clerical function.” So the court has stated that the requirements for patents (what they call the *enablement requirement*) are not much more than you can find by poking at a copy of the program for a while.

Mann and Sager [2005] interviewed software technologists and found that the apathy toward learning from patents revealed by the general surveys is evident in software as well: “. . . *none* of the startup firms to which I spoke suggested a practice of doing [patent] searches before beginning development of their products.” [p 1004, italics in original.]

And gosh, have you ever looked at a patent? It is a terrible way to teach others. I’ve read many a debate where some people having ordinary skill in the art look at a patent and declare it obvious and generally idiotic. Eventually, a patent examiner pipes up that the document is written in a technical language that laypeople outside the patent field are not trained to read. Looking at any patent, you can see that this is definitely true. It’d be difficult indeed to have a document that is both a legal declaration of a limited monopoly and a technical document for teaching fellow engineers. The teaching part takes a definite backseat to the legal declaration.<sup>4</sup>

So the disclosure story is an interesting theoretical anomaly: it makes intuitive sense on a broad scale, but we know both from looking around and from careful analysis of survey data that the broad intuition just doesn’t play out here in the real world. And yet people keep referring to it, because the broad concept makes sense.

So I won’t mention disclosure further, but will stick to the other, actually functioning, stories for why patents exist.

## References

Ashish Arora, Marco Ceccagnoli, and Wesley M Cohen. R&D and the patent premium. Technical report, 2003. NBER Working Paper No. 9431.

Anthony Arundel. Patents in the knowledge-based economy. *Beleidstudies Technology Economie*, 37:67–88, 2001.

Martin Campbell-Kelly. Not all bad: An historical perspective on software patents. *Michigan Telecommunications Technology Law Review*, 11(191), 2005.

---

<sup>3</sup>908 F.2d 931, 940–41 (1990).

<sup>4</sup>It is possible that after a patent is granted, the author then proceeds to produce a journal article. But you need a story beyond economics for why the journal article is getting written. No patent is a perfect monopoly, so the more information you give your competitor, the more power they have to invent around your work. So there are stories where disclosure plus journal publication work, but they partially advocate against the entire concept of patents (because the typical inventor really wants to show off and score publications) and I’ve seen no evidence that they dwarf the lack of evidence of benefits from disclosure up there in the main text.

Ronald J. Mann and Thomas Sager. Patents, venture capital, and software start-ups. *University of Texas Law, Law and Econ Research Paper*, (57), 2005.

William Thomson. *A Guide for the Young Economist: Writing and Speaking Effectively about Economics*. MIT Press, 2001.

Terry A Welch. A technique for high performance data compression. *Computer*, June 1984.