1 July 2009

Submission on the Patents Bill

To the Commerce Committee

This is a submission from Guy Burgess, Auckland, on the Patents Bill. *[edited]*

I am a solicitor practising in New Zealand (specialising in Information Technology law) and have a background as a software developer. I make this submission as an individual.

Introduction

Dear Committee Members

I request that the Committee amend the Patents Bill to **exclude** the patentability of all software, or methods implemented in software (together referred to as "software patents"). I have proposed a specific amendment to the Bill for your consideration (page 4).

For the reasons set out below, I submit that this is an issue of critical importance. I am aware that the Ministry and Cabinet have previously heard submissions on software patents, however with respect the Cabinet Paper did not address the matter in depth. In the October 2008 *New Zealand Law Journal*, Justin Graham (Principal, Chapman Tripp lawyers) stated:

> The Cabinet Paper acknowledged that business method and computer software patents are controversial, yet did not delve deeply into why this is so or provide guidance as to when these products would be patentable. The current approach effectively puts the business methods and computer software patents issue in the "too hard" or "leave it to the judges" basket.

I concur with Mr Graham's view (though I note this submission is limited to software patents, not business method patents). This issue is too important for the Committee *not* to undertake a thorough investigation into it. Once passed, the new Patents Act will likely be in place for many years. I therefore urge the Committee to consider this issue in detail.

My submission is that software patents should be excluded from patentability on the following grounds:

1. They unduly inhibit innovation, contrary to the intent and purpose of Bill.
2. Software is not a "manner of manufacture" but rather a "manner of configuration".

These grounds are set out below.

1. Software patents unduly inhibit innovation

Software patents operate to limit the possible uses of an infinitely configurable device – the computer.

Software implements processes and functions to manipulate and configure a machine (i.e. the computer) to produce a desired result. With modern computers and programming languages, there are almost no limits to the functions and processes which can be implemented by software (within the underlying capabilities of the computer). Within the machine, there are no physical constraints as to how the functions and processes may potentially be arranged, integrated, or otherwise used.

The result is an unfettered ability to innovate. It is possible for virtually any process to be implemented within a computer in some form. Among the most graphic examples of this are the "virtual worlds" such as Second Life, which create virtual physical environments, social communities and economies within computers. It is fair to say that today, if something can be imagined, it can be can created, in some form, by software.

This new, unfettered ability to innovate should not be diminished – potentially at a fundamental level and with the risk of legal liability – by the fact that someone else has patented what is essentially the configuration of a computer in a similar way. A software patent means that no one else is allowed to configure a computer in a similar manner, no matter how important that configuration may be for a particular desired use, without the patent owner's permission.

The result is that software patents are the **only** thing that can potentially inhibit innovation in the configuration of computers. In my view, this is contrary to the rationale of the Patents Bill, and as such software patents should be excluded from patentability.

There is no evidence to suggest that there is (or has been) any lack of innovation in software development, or that excluding software patents would lessen incentives for innovation. To the contrary, software has been part and parcel of the most important innovations of the past fifty years – the computer and internet revolution. That innovation is continuing apace. As noted above, the only threat to that innovation is the prevention of certain configurations via software patents. There have been local examples of software patents being used to attempt to prevent software developers and business owners from configuring computers in certain ways, clearly for illegitimate, anti-competitive purposes.[1] While these situations have been limited in New Zealand to date, they are more prevalent overseas and represent a real threat to software developers and users who – for commercial, educational or personal use, and knowingly or unknowingly – wish to configure a computer in a similar way.

The rationale of granting a monopoly – to encourage innovation – is redundant in the case of software development. The extension of monopoly protection to software patents (which will continue to be available if software patents are not excluded) is simply unnecessary. Monopoly protection should not be available without a clear need for it, in particular when granting it inhibits the usefulness and capacity for innovation using today's key foundational technologies – computers and software.

1 See for example: http://www.nzherald.co.nz/business/news/article.cfm?c_id=3&objectid=3511839

There is already very strong protection for software under copyright law. In most countries, software receives copyright protection (New Zealand's Copyright Act 1994 defines "literary work" to expressly include "computer programs"). Unlike a patent, copyright arises *automatically* in original works (such as software) without the need for registration and is internationally enforceable as of right. Copyright already protects a specific implementation of software, as well as other aspects such as "look and feel". Copyright protection has been proven to provide strong incentives for innovation. If software continues to be patentable, it means that software would be one of the only forms of invention capable of receiving the "double protection" of both copyright law *and* patent law. The addition of patent protection for software that is already automatically protected under copyright law is unnecessary, unwarranted, and unduly restrictive on innovation.

2. Software is not a "manner of manufacture"

Software is not a "manner of manufacture", which is the stated basis of patentability (Patents Bill clause 13(a)). Instead, it is a manner of **configuration**.

While there is no formal definition of "software", it is, at its most elementary level, a set of instructions for controlling a machine. The instructions that comprise the software are a finite set of methods or functions implemented by the machine or other software. The source code of software consists of text and numbers written in a particular computer language – the basis on which the Copyright Act 1994 (and other copyright acts around the world) expressly recognise software ("computer programs") as *literary works*.

Typically, the "machine" on which software runs is tangible hardware such as a desktop computer or a pocket calculator. However, the machine can itself be a "virtual machine" implemented entirely in software – other software running on the virtual machine behaves the same way as if it were running on a physical machine. Ultimately, the software **configures** how the machine runs. The software is simply a sequence of instructions written (as source code) in a particular computer language. It is comparable to a book, or a musical score, or a dance choreography – none of which are patentable (nor should be), nor considered "manners of manufacture". They are, however, protected by copyright. Software should be treated the same way.

On that basis, a software patent cannot be properly said to describe a "manner of manufacture" within the ordinary meaning of those words, and certainly not "within the meaning of section 6 of the Statute of Monopolies [1623]" as required by the Patents Bill clause 13(a).

It is acknowledged that the *process of producing* software may be described as a "manner of manufacture", and therefore makers of software could be described as "software manufacturers". It is possible – and within common concepts of patentability – that an inventor could apply to patent a "manner of manufacturing software", provided it met the required criteria. However, any resulting software – which is a manner of **configuration** – should not itself be described as a "manner of manufacture".

While I submit that software does not fall within the definition of "manner of manufacture"

in clause 13(a) of the Patents Bill, I submit that, if the Committee agrees, it should not be left to the Courts or the Patent Office to potentially arrive at contrary views, or to grant patents that unintentionally have that effect (as appears to have been the case in some overseas situations). If software patents are to be excluded, the Bill should expressly state the exclusion.

Proposed legislative change

The legislative change to the Patents Bill to exclude software patents would be very minor and have no impact on the rest of the Bill. I propose the following amendment to the Bill for the Committee's consideration:

> The addition of a sub-paragraph (6) to clause 15 as follows: **"15 (6) An invention is not patentable to the extent it is implemented in software."**

This simple amendment would *not* prevent the patenting of any invention involving or incorporating software as a component of the invention. It would simply mean that parts of an invention that are implemented in software could not form part of the patent claim. It would *not* prevent a patent for a process of *producing* software. The proposed amendment would also *not* prevent business method patents – although it would have the (desired) effect of preventing a person from obtaining such a patent where the business method is *entirely* implemented in software.

Conclusion

Software patents are incompatible with the stated intentions and objectives of the Patents Bill, and incompatible with commonly understood concepts of what *should be* patentable. Software is a configuration, and akin to a literary work – something that should not be considered a "manner of manufacture" or otherwise patentable. Through the workings of Courts and patent offices in some jurisdictions including New Zealand, patent law has been allowed extend to software in recent years. This has occurred without express legislative authority and due policy consideration. On that basis, although the issue has been raised on previous reviews of the Bill, I respectfully request that the Committee consider this important issue in due detail.

Yours faithfully,

Guy Burgess, LLB BCom (Information Systems) AMNZCS
Solicitor
guy@burgess.co.nz

1 July 2009