Dr Mark A Summerfield

SUBMISSION TO PRODUCTIVITY COMMISSION

Intellectual Property Arrangements – Draft Report

3 June 2016

CONTENTS

1. Introduction and Summary 1
2. My Interest and Qualifications 3
3. Recommendation on 'Business Methods and Software' 4
4. Incompatibility With TRIPS 4
5. 'BM&S' is Not Useful as a Category 5
6. Business Innovations Are Already Unpatentable 7
7. 'Open Source' Arguments Are Misplaced 8
8. 'Embedded Software' Cannot Be Clearly Defined 11
9. A Software Exclusion Will Not Exclude Software 13
10. Conclusion 15

1. INTRODUCTION AND SUMMARY

This submission is made in response to the invitation from the Productivity Commission to provide comment on the Draft Report on Intellectual Property Arrangements ('*Draft Report*'). . I make this submission on my own behalf, and based upon my background and experience as an electrical engineer, software developer, inventor and patent attorney.

My submission is directed to Chapter 8 of the *Draft Report*, and particularly to the draft recommendation that 'the Australian Government should amend section 18 of the *Patents Act 1990* (Cth) to explicitly exclude business methods and software from being patentable subject matter'. A short summary follows.

- The proposed exclusion is incompatible with Australia's obligation under the *Agreement on Trade-Related Aspects of Intellectual Property Rights*.

- The category of 'business methods and software' ('BM&S') developed in the *Draft Report* does not provide a sufficiently nuanced view of the wide diversity of existing and future computer-implemented products, systems and services. Implementing an exclusion from patentability on this basis would result in unpredictable and unintended consequences, and would discriminate against innovators whose use of programmable hardware and software to create new technology solutions would be denied patent protection while innovations based upon more traditional technologies would continue to enjoy the benefits of the patent system.

- Following the recent decision of the High Court of Australia to deny special leave to appeal the decision of the Full Court of the Federal Court of Australia in the *RPL Central* case, the law in this area is now settled. There is a clear distinction between technological innovations, which are patentable, and business innovations, which are not. An express exclusion for business methods is therefore unnecessary, and its introduction would disrupt what is now a stable position in the law.

- Open source models are complementary to proprietary software development, with many small and large software developers now working with a mix of open source and proprietary software and platforms. The existence of open source development models provides little insight into the role played by IP protections, including patents, within software-based industries.

- The experience of those countries, including the UK, in which patent protections for computer-implemented inventions have been 'wound back',

suggests that very little may change in the long term if the exclusion proposed in the *Draft Report* were to be implemented. The main effect would therefore be to create new uncertainty just as the law appears to have been settled in a manner that is broadly consistent with the position of a number of Australia's major trading partners.

Overall, it is my submission that the existing 'manner of manufacture' test, as it has been developed in the recent decisions of the Full Federal Court, strikes an appropriate balance, and that no change is necessary.

2. MY INTEREST AND QUALIFICATIONS

I have a Bachelor's Degree in Electrical and Electronic Engineering, and a PhD in optical fibre technology, both earned at the University of Melbourne.

I completed my undergraduate degree at the end of 1989, and during 1990 and 1991 I worked at the Research Laboratories of Telecom Australia (now Telstra Limited).

Between 1992 and 1995 I was involved in experimental and theoretical research within the Photonics Research Laboratory (PRL) at the University of Melbourne relating to optical fibre communications systems, leading to the award of my PhD in 1996.

I subsequently worked in a number of research and development, and intellectual property management roles, including three years with two separate high-tech start-up companies developing and commercialising technology spun-out of university research programs.

I am the first-named inventor on US Patent No. 7,233,962, *Optical Error Simulation System*, which was issued on 19 June 2007.

In late 2002 I commenced as a trainee patent and trade marks attorney with Watermark Patent & Trade Marks Attorneys ('Watermark') in Melbourne, Australia. In 2005 I was first registered as a Patent and Trade Marks Attorney in Australia and New Zealand.

I am currently Special Counsel with Watermark, and the editor and primary author of the *Patentology* blog[1], which covers current issues relevant to Australian and New Zealand patent applicants and practitioners. I am also a member of the Institute of Electrical and Electronic Engineers (IEEE)[2]. Much of my day-to-day practice involves assisting clients with the management and protection of intellectual assets relating to information technologies, including telecommunications technologies and software-implemented inventions. I drafted the patent specification at issue in the *RPL Central* series of cases, and was involved throughout the stages of examination, opposition, Federal Court (first instance) appeal, and Federal Court (Full Court) appeal.

My interest in Australia's intellectual property system, and the patent system in particular, is not only as an advisor to my current clients, but is also based on my

[1] http://blog.patentology.com.au

[2] http://www.ieee.org

past experience in the development, protection and commercialisation of innovative new technologies, many of which have been implemented wholly, or in part, via software.

3. RECOMMENDATION ON 'BUSINESS METHODS AND SOFTWARE'

Chapter 8 of the *Draft Report* is devoted to 'business methods and software patents'. The Commission adopts the abbreviation 'BM&S' for 'business methods and software', and uses this abbreviation throughout the chapter, thereby aggregating a broad range of technical and non-technical subject matter into a single homogenous category.

Draft Recommendation 8.1 is that 'the Australian Government should amend section 18 of the *Patents Act 1990* (Cth) to explicitly exclude business methods and software from being patentable subject matter.'

I strongly oppose the introduction of such an exclusion. Indeed, I consider that it would represent the very epitome of the phrase 'throwing the baby out with the bathwater.' Reading Chapter 8 of the *Draft Report*, I am left with the distinct impression that the Commission has a very limited conception of the nature and extent of the role played by computer-implemented innovation across a broad range of industries.

Aside from 'embedded software' (or, as described on page 252 of the *Draft Report*, 'software embedded in inventions'), the Commission appears to regard all software as broadly comparable with the computer programs of our everyday experience as workers and consumers, e.g. office productivity suites, web browsers, and the 'client-facing' components of web-based systems and services such as e-commerce sites. However, this results in a very narrow view of the complexity, development cycle, risk profile and cost of much important and innovative software that underpins systems and services that we have come to take for granted.

4. INCOMPATIBILITY WITH TRIPS

As the Commission is aware, Article 27(1) of the *Agreement on Trade-Related Aspects of Intellectual Property Rights* ('TRIPS') provides that:

> *Subject to the provisions of paragraphs 2 and 3 [which have no bearing on computer-implemented inventions], patents shall be available for* ***any inventions, whether products or processes, in all fields of technology****, provided that they are new, involve an inventive step and are capable of industrial application.*

As a member of the World Trade Organization (WTO), and therefore a signatory to TRIPS, Australia is not at liberty to exclude from patentability subject matter that includes products or processes in any 'field of technology'. A blanket exclusion of 'software', as is proposed in Recommendation 8.1, is clearly incompatible with Australia's obligations under TRIPS.

There is no question that many computer-implemented inventions (i.e. 'any invention the performance of which involves the use of a computer, computer network, or other programmable apparatus, the invention having one or more features which are realised wholly or partly by means of a computer program or computer programs'[3]) are technical in nature, and are thus entitled to patent protection under TRIPS.

The 'technical' nature of many computer-implemented inventions is addressed further in a number of the following sections of this submission.

5. 'BM&S' IS NOT USEFUL AS A CATEGORY

The *Draft Report* defines a business method as 'a way of operating "any aspect of an economic enterprise", and encompass[es] a broad range of ideas and activities'; it defines software as 'a set of instructions that allow computing devices to function' (pages 233-234). The *Draft Report* further states that 'with the rise of the digital economy, many business methods are implemented by software, making it difficult to separate the two' ('Key Points', page 233).

With all due respect to the efforts made in the *Draft Report* to tackle complex issues in both technology and patent law, this is a gross simplification that has virtually no basis in reality. In my experience, 'BM&S' is an invented and substantially non-existent category of subject matter, the primary effect of which is to conflate subject matter as diverse as, at one extreme, asset protection schemes and methods for generating investment portfolio indexes that have been found to be unpatentable and, at another extreme, highly technical and mission-critical software used to control industrial manufacturing processes.

The *Draft Report* may be correct to say that 'many business methods are implemented by software.' At the very least, many business processes are supported by software, even if it is just through the use of common business productivity tools such as spreadsheets, databases, word processors, and project management applications.

---

[3] This definition is taken from the *Proposed Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions (2005)* which, while ultimately not adopted , remains in my view the most workable expression of the subject matter at issue.

However, it is equally self-evident that the majority of such processes employed within business across Australia, and around the world, whether or not supported by software, are not the subject of patents, or indeed of any form of IP protection other than, perhaps, trade secrecy.

It is also apparent that the vast majority of software is not directed to implementing business methods. The software that controls a car's engine, a washing machine, a dishwasher, the Wi-Fi implementation in a mobile phone, the complex systems of an Airbus A380, a wireless router, the digital video decoders in a television, a set-top box, or a Blu-ray player, the switching equipment in the telecommunications networks, life support systems, X-ray machines, gene sequencers, systems on the International Space Station, satellites, warships, military drones and guided weapons, public transport signalling systems, traffic management systems, security systems, and laboratory test and measurement apparatus are all examples, among countless others, of software that provides functionality completely unrelated to the performance of any business method. All such software either enables entirely new functionality, or replaces prior electronic and/or mechanical systems with a cheaper, more efficient and/or more powerful digital alternative.

While the *Draft Report* recognises the role of what it calls 'embedded' software, it pays only lip service to this subcategory, as if it is some sort of minor exception to a general rule that software (as subsumed within the 'BM&S' category) is not deserving of patent protection. In reality, programmable hardware and software are 'embedded', out of plain view, throughout modern society, in everything from small domestic appliances such as toasters to the huge global networks of server and terminal systems that manage international air transport operations processing bookings, flights, baggage handling, security and so forth for millions of passengers daily.

However, there are also numerous software applications that run on desktop computers, and other common computing devices, that are highly technical in nature, and do not implement anything vaguely resembling a 'business method'. For example, software that assists in the design, simulation, construction and/or manufacturing of infrastructure (e.g. buildings and bridges), integrated circuits, electronic systems, mechanical devices, communications networks, aerospace systems and so forth (i.e. computer aided design and manufacturing, or 'CAD/CAM', software) is highly technical in nature, and is based upon principles of physics and engineering, not business.

Furthermore, within the realm of computer systems as technical artefacts in their own right, there is software that manages huge databases, performs highly complex information processing, that performs data compression, coding,

decoding, and other forms of manipulation, and which is based on advanced principles developed in the fields of mathematics, information theory, and computer systems engineering, sometimes at significant expense and considerable commercial risk.

The fact is that the vast majority of important code in the world – code that requires large investments in research, development, coding and testing – is not to be found in smartphone apps or on e-commerce sites. This code, whether embedded in products and systems, or used for their design and operations, is mission-critical, mostly proprietary, and carries with it huge liability in case of faults or failures.

To sum up: most business processes are not critically dependent upon software, and most software does not implement any business process. The category of 'BM&S' developed in the *Draft Report* does not provide a sufficiently nuanced view of the wide diversity of existing and future computer-implemented products, systems and services.

Any legislative change based upon such a broad and artificial class of subject matter will lead to unintended consequences. It would discriminate against businesses whose use of programmable hardware and software to create new and innovative technologies and solutions unrelated to the performance of mere business processes would be denied patent protection while innovations based upon more traditional technologies would continue to enjoy the benefits of the patent system.

6. BUSINESS INNOVATIONS ARE ALREADY UNPATENTABLE

The *Draft Report* makes a number of references to the *RPL Central* case.[4] In particular, at page 241 it notes that the 'consensus' on patenting of business methods in Australia may be 'short lived', citing the application by RPL Central Pty Ltd for Special Leave to appeal the Full Court decision to the High Court of Australia. On page 251 it describes *RPL Central* as 'a case unsettled'.

This is no longer the case. On 5 May 2016, the High Court denied the application for Special Leave, stating simply that 'the Full Court was plainly correct and, accordingly, none of the applicant's proposed grounds of appeal enjoys sufficient prospects of success to warrant the grant of special leave to appeal.'

[4] *Myall Australia Pty Ltd v RPL Central Pty Ltd* [2011] APO 48; *RPL Central Pty Ltd v Commissioner of Patents* [2013] FCA 871; *Commissioner of Patents v RPL Central Pty Ltd* [2015] FCAFC 177.

Over three recent decisions (*Grant*[5], *Research Affiliates*[6] and *RPL Central*[7]) the Full Federal Court has consistently confirmed that there 'is a distinction between a technological innovation which is patentable and a business innovation which is not'.

Further, in *RPL Central*, at [96], the Full Court stated plainly that 'a business method, or mere scheme, is not, per se, patentable'. In a case in which the claimed invention appears to be directed to a business method or scheme, it is 'necessary to understand where the inventiveness or ingenuity is said to lie' (at [112]). For the invention to be patentable, there must be ingenuity in the technical implementation.

It is therefore now clear and settled that business methods are not patentable in Australia. Technological inventions may, however, be patentable, in which case no distinction is drawn between whether the invention is (or may be) implemented via software and/or other technical means, or to the particular field of economic utility in which the invention is applied.

To introduce an express exclusion for 'BM&S', in circumstances in which business methods are already unpatentable under the 'manner of manufacture' test, and in which a significant proportion of computer-implemented inventions are plainly technical in nature, and thus entitled to patent protection in accordance with TRIPS, would serve only to create confusion and uncertainty.

It is likely that it would be many years before a case addressing the interpretation of such legislation would reach the superior courts (i.e. the Full Federal Court and/or the High Court). It would therefore be damaging and disruptive to introduce such a substantial change to the Australian patent law at a time when the position seems finally to have been settled.

7. 'OPEN SOURCE' ARGUMENTS ARE MISPLACED

The *Draft Report* appears to place some weight on the existence and success (in some areas, at least) of 'open source' software. It has become commonplace, whenever there is a review of patentable subject matter, for supporters of the free and open source (FOSS) software movement to lobby aggressively for inventions involving software to be denied the IP protections afforded to other technological innovations.

---

[5] *Grant v Commissioner of Patents* [2006] FCAFC 120 at [24].

[6] *Research Affiliates v Commissioner of Patents* [2014] FCAFC 150 at [93].

[7] [2015] FCAFC 177 at [100].

The ‘open source’ argument essentially consists of asserting that the existence of successful FOSS projects – such as the Linux operating system – and the increasing embrace of open source practices by commercial software developers, such as Google, Microsoft, IBM and Apple, is somehow evidence that the ‘software industry’ (assuming such a monolithic entity actually exists) would continue to invest and to innovate in the absence of patent protection, and even that proprietary models of software development are becoming obsolete in view of the wider benefits of open source development.

The influence of this argument is evident in the *Draft Report*, which states (page 244, with citations omitted):

> *In the case of software developments, many owners do not protect their content at all. Indeed, rather than seek protection some developers share their code and encourage third parties to copy and contribute to the development of their software. This approach, referred to as open source, has been adopted by companies including Google, IBM, RedHat and Sony.*
>
> *Open source approaches offer a number of benefits for both software developers and innovating firms. Developers derive a gain from making software, as it allows them to learn new skills from collaborators. Firms gain as open source approaches are typically more agile and adaptable and can be built to accommodate follow on innovations without the need for proprietary products, allowing for fast and dynamic improvements.*

Open source software is not an ‘alternative’ to proprietary software, as is amply demonstrated by its embrace by companies such as IBM and Microsoft, which rely upon proprietary products (which may or may not incorporate and/or contribute open source components) for their revenues. Notably, these companies are also among the world’s largest users of the patent system as one means of protecting their large investments in research and development.

Furthermore, patent and other intellectual property rights are entirely compatible with open source innovation and, indeed, some of the most successful open source models rely heavily upon copyright protection to support their objectives. For example, the ‘GNU General Public License’ (GPL) promoted by the Free Software Foundation, employs so-called ‘copyleft’ terms that require anyone distributing software that incorporates covered open source code (whether or not it has been modified) also to make the corresponding source code available openly on the same licence terms. The GPL open source model is entirely dependent upon IP rights. If the original author of the code had no such rights, there would be no legal basis for the licence terms to be enforced, and thus no

way to compel downstream users of the open source code to participate in the open source movement.

Other open source licences, such as the 'BSD licence' originating at the University of California, Berkeley, and the 'Apache Licence', which covers many open source projects including the world's most widely-deployed web server platform, are less restrictive, and do not require downstream users to publish the source code of proprietary products built using covered open source components.

Most open source projects comprise components that may be reused in larger software systems, i.e. they are building blocks that may be assembled and 'glued together' with new (and possibly proprietary) code in the development of new software products and services. While there are a number of notable open source products that can be installed and used by end consumers with little or no specialised technical knowledge (e.g. the Linux operating system, the LibreOffice office productivity software, the Firefox web browser and the Thunderbird email client application), these represent a minority of the total open source code available today.

My own view is therefore that proprietary and open source models are complementary to one another, and these days much software development lies somewhere along a continuum involving a mix of open source and proprietary technologies. Each has its place.

While Microsoft has its detractors, it has for decades now been delivering consumer- and business-grade software that has immeasurably enhanced the productivity of people throughout the world. With Windows NT and Windows 95, Microsoft brought full 32-bit multi-user, multi-processing operating systems to the masses by the mid-1990s, when Linux was still too unstable to be of any use to anybody but those working on its development. The same can said for many other proprietary applications that continue to deliver the levels of usability and stability demanded by businesses, and most consumers, where the FOSS alternatives still require technical expertise, self-support skills and patience that most users do not possess. I would count among these Adobe Photoshop, Microsoft Office, and all of Apple's proprietary software.

Microsoft spent over US$11 billion on research and development in 2015, and was granted 1956 US patents. IBM invested US$5.2 billion, and received 7355 US patents. Google's R&D spend was US$9.8 billion, with a haul of 2835 US patents. Apple spent US$6 billion, and was granted 1938 US patents.[8] These

[8] R&D figures from statistica.com (http://www.statista.com/), and patent numbers from IFI CLAIMS (http://www.ificlaims.com/index.php?page=misc_top_50_2015).

companies, among others, are conducting quality research and development of new technologies and, accordingly, their patents are not, for the most part, directed to business methods or other non-technical matter.

Conversely, their size, depth of market penetration, and ability to subsume innovations developed by smaller companies and individuals (whether open source or otherwise) is, in the absence of patent protection for those innovations, essentially unrestricted. What is a small business, providing a successful add-in for a Microsoft product, supposed to do if and when Microsoft decides to deliver the same functionality to all of its customers worldwide as part of a routine automated update? Without effective IP protection there is nothing it can do.

In short, open source plays an important role in modern software industries, but it is not evidence that all software innovation and (importantly) successful commercialisation would occur in the absence of IP protections, including patents for technological innovations based on programmable hardware and software.

8. 'EMBEDDED SOFTWARE' CANNOT BE CLEARLY DEFINED

In Information Request 8.1, the Commission has expressly requested input on whether there are any viable tests for distinguishing between patent-worthy and unworthy examples of 'embedded software'. This request is, in my view, founded upon a serious category error, and as such has no satisfactory response.

The *Draft Report* refers to the New Zealand experience in this regard[9], and yet apparently fails to learn anything from it. I agree completely with the words of the New Zealand Ministry of Economic Development:

> *There is no simple definition which exactly captures the idea of 'embedded software' or 'embedded systems'. Any attempt to provide a simple definition may make it relatively easy for patent attorneys and applicants to use 'creative drafting' to avoid the definition. Devising a simple definition is likely to be difficult, if not impossible. Technical advances may mean that any definition fixed in legislation becomes obsolete fairly quickly.*

A contemporary example of the truth of this statement may be found in mobile technology. In the early days of digital cellular mobile telephony, the software executing on consumer mobile devices was widely regarded by electrical

[9] Box 8.1, page 250, 'Starting hard, going sheepish?' If I may offer this additional feedback on the draft, I suggest that if this box is retained in the final report it should be retitled. The title is offensive to our New Zealand neighbours and (though perhaps unintentionally) a little obscene.

engineers as 'embedded'. It was generally proprietary, and highly customised to the specialised hardware on which it was executed.

More recently, however, the rise of smartphones, tablets and other mobile devices has completely blurred the boundary between embedded systems and conventional 'nonembedded' systems, such as general purpose desktop PCs. There has been a degree of standardisation in the hardware, and operating systems such as Apple's iOS and Google's Android (which itself is built on the open source Linux operating system) have provided further standardisation in programming interfaces that have enabled developers to build applications ('apps') that can be installed by end users to customise the functionality of their mobile devices, just as they do with their desktop systems.

A February 2012 feature article in Electronic Engineering Journal[10] addressed the question of whether mobile devices are embedded systems. Aside from noting that even experienced engineers can (and do) argue for hours over the answer to the question 'what is an embedded system?', and discussing possible definitions for embedded systems, the article notes that:

> *…both smartphones and tablets violate the use definition of an embedded system. But they're also not general-purpose computers. They're something in between. Validating mobile as a separate category.*

Given that experienced electronics engineers working in relevant fields cannot agree on a definition for embedded systems, let alone subcategories within such a category, the incorporation of a suitable definition into either the patent law, or the practice of the Patent Office, is plainly unworkable. As the New Zealand Ministry of Economic Development quite rightly went on to say:

> *No other country has attempted to make the distinction between 'embedded' and 'nonembedded' computer programs in patent legislation. Implementing such a distinction in New Zealand is problematic, as IPONZ would not be able to make use of case law or practice developed elsewhere. IPONZ and the New Zealand courts would have to develop their own practice, from scratch, with decisions from other jurisdictions providing little, if any, guidance. It may take some years, and a number of court cases, to develop a consistent and coherent practice that provides certainty to all concerned. There would be considerable uncertainty for both IPONZ and patent applicants.*

---

[10] Bryon Moyer, 'Are Mobile Devices Embedded Systems?' Electronic Engineering Journal, 27 February 2012, http://eejournal.com/archives/articles/20120227-mobile/, retrieved 3 June 2016.

This would be equally true of any similar attempt in Australia.

9. A SOFTWARE EXCLUSION WILL NOT EXCLUDE SOFTWARE

As the *Draft Report* notes (at page 249) a number of countries have 'wound back' patent protections for computer-implemented inventions, with the UK being mentioned among those that have legislated exclusions. However, this has not resulted in all computer-implemented inventions being excluded from patentability.

Recent developments in the UK are instructive as to how 'computer program' exclusions have been interpreted in a manner that is consistent with the TRIPS obligation to make patents available in all fields of technology, as well as illustrating the kinds of computer-implemented technological innovations that remain patentable.

Subsection 1(2) of the UK *Patents Act 1977* implements the UK's obligation to conform its law to Article 52(2) and (3) of the European Patent Convention (EPC), and relevantly provides that:

> *(2) It is hereby declared that the following (among other things) are not inventions for the purposes of this Act, that is to say, anything which consists of –*
>
> *...*
>
> *(c) a scheme, rule or method for performing a mental act, playing a game or doing business, or* ***a program for a computer****;*
>
> *...*
>
> *but the foregoing provision shall prevent anything from being treated as an invention for the purposes of this Act only to the extent that a patent or application for a patent relates to that thing* ***as such****.*

In the *Symbian* case[11], the UK Court of Appeal stated (at [48]):

> *We turn to address the issue whether the Application in this case was excluded from registration on the ground that it was a "program for a computer … as such". The mere fact that what is sought to be registered is a computer program is plainly not determinative. Given that the Application seeks to register a computer program, the issue*

[11] *Symbian Ltd v Comptroller General of Patents* [2008] EWCA Civ 1066. The patent application was directed to a method of accessing data in a dynamic link library in a computing device.

> *has to be resolved by answering the question whether it reveals a "technical" contribution to the state of the art.*

The *Symbian* decision was subsequently relied upon by Judge Birss in the UK High Court (Patents) in the *Halliburton* case.[12] *Halliburton* was concerned with patent claims directed to improving the design of roller cone drill bits for drilling oil wells (and the like). The claimed invention employed computer simulation of the interaction of the drill bit with the material being drilled to optimise various design features of the drill bits. The use of computer simulation was intended to reduce or eliminate extensive field testing. The output of the claimed computer-implemented method was a specification that could be used to manufacture the designed drill bit. However, actual production of the drill bit did not form part of the claimed invention.

Judge Birss applied the four-step test for assessing patent-eligibility set out by the Court of Appeal in *Aerotel/Macrossan*[13], and found that the invention was patentable on the basis that:

> *Designing drill bits is obviously a highly technical process, capable of being applied industrially. Drill bit designers are, I am sure, highly skilled engineers. The detailed problems to be solved with wear and ability to cut rock and so on are technical problems with technical solutions. Accordingly finding a better way of designing drilling bits in general is itself a technical problem. This invention is a better way of carrying that out. Moreover the detailed way in which this method works - the use of finite element analysis - is also highly technical. (At [74].)*

The point of all this is to demonstrate that a computer-implemented invention making a substantive technical contribution to the relevant field, such as an improvement in computer performance/reliability (*Symbian*) or improvements in engineering design capability (*Halliburton*), remains patentable in the UK despite the implementation of a statutory prohibition on the patenting of 'a computer program … as such.'

This is consistent with the UK's TRIPS obligation to make patents available in all fields of technology. It is also consistent with the position in Australia, according to recent authority of the Full Federal Court in *Grant*, *Research Affiliates* and *RPL Central*, that a technological innovation is patentable (regardless of whether or not its implementation may involve software), while a business innovation, or a mere scheme is not.

---

[12] *Re Halliburton Energy Services Inc* [2011] EWHC 2508 (Pat).

[13] *Aerotel v Telco / Macrossan's Application* [2006] EWCA Civ 1371.

Any attempt, at this time, to amend the *Patents Act* to introduce some form of express exclusion for 'business methods and software' would serve only to create new uncertainty at a time when the law seems finally to be settled in a way that is broadly consistent with the position in countries with which Australia conducts significant trade, including the UK (and other European countries covered by the EPC), the US, and countries such as China in which the patent laws are closely modelled on the European laws.

10. CONCLUSION

I do not support the draft recommendation that section 18 of the *Patents Act 1990* (Cth) be amended to explicitly exclude business methods and software from being patentable subject matter.

For the reasons discussed in detail above, I believe that such a change is unnecessary, would create new uncertainty at a time when the law in this area seems finally to be settled, and would not, when interpreted in a manner consistent with Australia's obligations under TRIPS, result in the exclusion of a significant class of computer-implemented inventions that are not already unpatentable in accordance with the authority of the Full Federal Court in *Grant*, *Research Affiliates* and *RPL Central*.

I consider that the existing 'manner of manufacture' test, as it has been developed in the recent decisions of the Full Federal Court, strikes an appropriate balance, and that no change is necessary.

Mark Summerfield
3 June 2016